VCU College of Engineering

CS 25-341 Cyber Intrusion Detection and Visualization
Project Proposal

Prepared for

Bjorn Vaagensmith

Idaho National Laboratory

By

Calvin Hurlbert

Christopher Malatesta

Jaspreet Singh

Di-Huy Tran

Under the supervision of

Milos Manic

Dec 9, 2024

Executive Summary

Problem Statement - Section A

This project plans to provide a solution identifying a Radio Frequency Identification (RFID) attack. This can be very difficult. RFID attacks leave almost no trace and have no way to identify that one was carried out until damage has been done. We wish to change this by developing a way to visually identify an attack.

Engineering Design Requirements - Section B

The requirements for this assignment are very loose which allows us to have a lot of freedom in the design of our visualizer but there are a few points that need to be made. This project needs to focus on anomaly detection and must be a visualizer. This design should stay below $1,000 in cost however if needed more can be acquired.

Scope of Work - Section C

The scope of our project is to develop visualization software that can detect and alert a user to anomalies found using tools like Raspberry Pi and Proxmox 3 This will all be done without our project's timeline using iterative development. This project will not cover attack mitigation past the detection of an anomaly on an RFID system.

Table of Contents

Section A. Problem Statement

This project plans to provide a solution to identifying a Radio Frequency Identification (RFID) attack. RFID attacks can be very difficult to detect since attacks usually leave no trace. We want to change this by developing a way that visually alerts victims to an attack on their RFID system and helps them easily reach a solution to protect their systems. This project also plans on testing how the visualization benefits an operator's ability to detect and react to an attack in real-time.

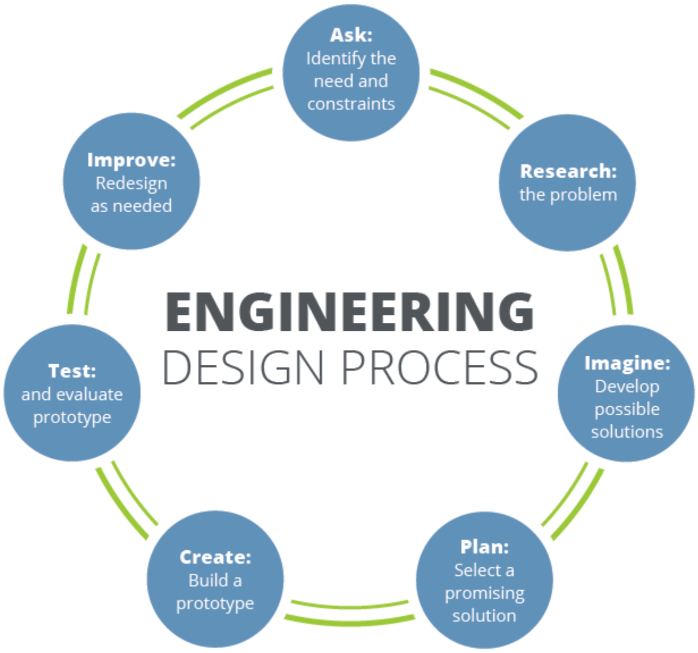

ENGINEERING
DESIGN PROCESS
Ask: Identify the need and constraints
Research: the problem
Imagine: Develop possible solutions
Plan: Select a promising solution
Create: Build a prototype
Test: and evaluate prototype
Improve: Redesign as needed

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

Section B. Engineering Design Requirements

The requirements for this assignment are very loose which allows us to have a lot of freedom in the design of our visualizer but there are a few points that need to be made. This project needs to focus on anomaly detection and must be a visualizer that is easy to read so that almost anyone can understand the situation. This design should stay below $1,000 in cost however if needed more can be acquired. For testing, we need to make a testbed and develop attacks and attack our testbed to see if our visualizer works. We have a few options for which types of attacks we could carry out varying from car keyfobs to RFID badges.

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

The goal of this project is to enhance cyber attack detection and response, particularly for RFID systems and networks, by developing an intuitive way to visualize and alert users of a potential threat. Specifically, the aim is:

- To create a user-friendly interface to represent network anomalies.
- To minimize the time between an onset attack and its detection.
- To better prepare systems for cyber-physical attacks and improve situational awareness for its administrators for a real-time response.

B.2 Design Objectives

The design will be able to identify anomalies, categorize them into three distinct threat levels, and suggest a course of action to respond to them.

- The design will include a user-friendly interface that is easy to understand and interact with with minimal training.
- The design will create a comprehensive report of network activity every 24 hours.
- The design will be able to handle high amounts of traffic data without degradation of performance.
- The test bed for the design must reflect a real-world environment for attacks.

B.3 Design Specifications and Constraints

- All aspects of the project must fit within the $1000 budget.
- The system must operate on a Raspberry Pi 4 with at least 4GB RAM, ensuring real-time processing capability within the hardware constraints.

- The design will provide a real-time visual representation of system activity, updating once every second.
- The interface must be easily understood by users who have no prior cybersecurity experience, within 10 minutes of interaction.
- The system must generate reports and log detected anomalies within 24 hours, and using no more than 500MB of storage space per session.
- The system must be able to handle at least three common RFID protocols.
- The simulated training environment must include at least 5 pre-programmed attacks without disrupting system performance.
- All data processed and stored by the system must be encrypted using AES-256 encryption
- The design must easily integrate into an existing network within 24 hours.

B.4 Codes and Standards

1. **NIST SP 800-53** - Security and Privacy Controls of Information Systems:
   - Offers guidelines for implementing security controls and safeguards in IT systems, critical for the Project's development on intrusion detection.
   - Provides a framework to ensure that the RFID system aligns with best practices in securing data against unauthorized access.
2. **FCC Part 15 -** Radio Frequency Devices:
   - This relates to the Proxmark3's usage as it involves radio frequency (RF) communication with RFID systems.
   - Ensures that the RF signals used for testing and analysis are within acceptable limits and do not interfere with other electronic devices or systems.
3. **ISO/IEC 27001** - Information Security Management Standards:
   - Relevant for ensuring the proper security management of data during RFID interactions and transmissions.
   - Helps define the requirements to secure data handling, encryption protocols, and risk management during the analysis and detention processes.
4. **IEEE 802.11 Standards** - Wireless LAN Standard:
   - Governs the wireless communication aspects of the project, particularly for devices like the Raspberry Pi that may connect wirelessly to during the testing phases.
   - Ensures interoperability and secure communication channels in scenarios where the project relies on wireless data transmission.
5. **OSHA 1910.147 -** Control of Hazardous Energy (Lockout / Tagout):
   - While primarily for physical safety, it is essential when working with hardware components like the Raspberry Pi and Proxmark3 to ensure safe handling during the testing and implementation phases.

Section C. Scope of Work

The scope of this project is to develop a robust system for detecting and visualizing cyber intrusions, specifically focusing on vulnerabilities and attacks targeting RFID systems. The primary objectives include analyzing RFID communication protocols, detecting unauthorized access or attacks using tools like Raspberry Pi and Proxmark3, are visualizing potential security breaches in a user-friendly interface.

- **Project Timeline**: The project will be conducted within a certain time frame, with milestones set for the completion of system design, hardware integration, intrusion detection programming, testing, and final report generation.
- **Methodology**: The team will use the part of Agile methodology for the approach, allowing for iteration testing and improvement based on the reviews and feedback from our representatives.
- **Responsibilities:** The team will be responsible for developing the intrusion detection system, testing with various RFID tags and frequencies, and creating a visualization dashboard. Tasks such as the setup of test environments, coding of detection algorithms and presentation materials will be managed by the team.
- **Exclusions:** The project will not cover comprehensive attack mitigation beyond detection nor focus on physical security aspects of RFID systems.
- **Verification and Approval:** Our weekly meetings and progress reports will be used for continuous feedback and validation of the project's progress.

C.1 Deliverables

- **Technical Deliverables:**
    - Implementation of RFID-based based intrusion detection system using Raspberry Pi and Proxmark3
    - Source code for detection algorithms and scripts for RFID analysis.
    - Functional dashboard for visualizing detected intrusions and potential attack vectors
    - Documentation covering the setup, operation, and testing procedures of the system.
    - Detailed reports on testing outcomes and analysis of RFID vulnerabilities.
- **Academic Deliverables:**
    - Team contract and project proposal submission.
    - Preliminary design report and project timeline.

    - Fall presentation and poster detailing project progress.
    - Final Design Report and Project timeline.
    - Final design report summarizing the entire project, including methodologies, results, and future recommendations.
    - Capstone EXPO poster and final presentation to show the project results.
- **Risk Mitigation:**
    - Access to the campus is required for hardware testing and final presentation setups.
    - Most coding and report writing can be completed remotely, utilizing shared drives and version control platforms like GitHub.
    - Hardware components have already been acquired to mitigate potential supply chain delays but any replacements or additional parts will be ordered well in advance.

C.2 Milestones

Project Proposal Tasks

1. Work on the proposal (09/26 - 10/11)
   This period involves the development of the overall project proposal. During this phase, we will be drafting the initial structure, and outlining project goals, design requirements, milestones, and things we'll need to work on the project throughout the semester.
2. Break down proposal (09/26)
   A one-day task where the team divvies up the project proposal into smaller, more manageable components for each team member to complete.
3. Finalize project idea (10/02)
   This milestone marks the point at which the core project idea was finalized. All key decisions and directions for the project should have been determined by this date.
4. Present idea to our advisor (10/03)
   The project idea was formally presented to our advisor in our weekly meeting for feedback and approval. This step ensures alignment with academic requirements and expectations, as well as ensuring everyone is on the same page for where the project is headed.
5. Review proposal (10/10)
   A final review of the project proposal is planned before submission or further progress. It's a chance for our advisor and company sponsor to refine, revise, and ensure everything is in order.

Fall Design Poster Tasks

1. Work on poster (10/12 - 11/15)
   This phase involves creating the design poster, which will visually communicate our project idea, design choices, and intended outcomes.
2. Break down poster (10/17)
   The poster content will be divided into key components to ensure a logical and organized layout, making the complex project information easily digestible. We will also be able to create a more detailed timeline and set of milestones for this section of the project.
3. Review poster (11/14)
   A review day is dedicated to checking the poster for accuracy, design quality, and overall clarity with the advisor before it is finalized.

Preliminary Design Report Tasks

1. Work on report (11/16 - 12/09)
   The preliminary design report will outline our project's theoretical framework, design decisions, and early results or projections. This phase is about drafting the document.
2. Break down report (11/21)
   A focused task to divide the report into structured sections, ensuring the content flows logically and covers all necessary details. We will also be able to create a more detailed timeline and set of milestones for this section of the project.
3. Review report (12/05)
   This milestone is for reviewing the full report draft, refining sections, addressing feedback, and ensuring it meets all the required criteria before submission.

Other Tasks

1. Order needed supplies (10/03 - 10/17)
   During this time, we'll be procuring any essential hardware, software, and other possible items for the project's technical setup.
2. Set up Raspberry Pi servers (After 10/17 for 1 week)
   Once supplies are ordered, this milestone involves setting up the Raspberry Pi servers, a key component of the project's infrastructure.
3. Set up Zeek on the server (After Raspberry Pi setup for 1 week)
   After the server is configured, Zeek (the Network Security Monitor) will be installed and set up for network monitoring purposes.
4. Run attacks on the server (After Zeek setup for 2 weeks)
   Following the Zeek setup, cyber-attacks will be simulated on the server to test the system's detection capabilities and gather data for analysis.
5. Design visual of attacks (After attacks for 2 weeks)
   Once all attacks have been run and detected, we'll create a visual representation of the

cyber-attacks to effectively communicate what our project entailed to those not well versed in cyber security.

C.3 Resources

3 Raspberry Pi Boards

The Raspberry Pi boards will act as small, versatile servers to simulate a network environment where we'll install Zeek and run cyber-attacks. These boards are ideal for lightweight network setups and can handle the necessary traffic monitoring and attack detection tasks. These will serve as the backbone of our project's cyber network infrastructure. We will use one Raspberry Pi to host the RFID server and the other run the attacks against the server.

2 Proxmark 3s

The Proxmark 3 devices will emulate and interact with RFID tags and NFC systems. These tools are essential for testing the security of physical access control systems and conducting various RFID-based attacks. Using the Proxmark 3s, we can simulate cloning, intercepting, and modifying RFID communications. This will allow us to evaluate the resilience of our system against unauthorized access attempts. The Proxmark 3s will also provide a hands-on approach to exploring how different RFID protocols can be exploited, giving us valuable insight into potential vulnerabilities and defense mechanisms.

2 Flipper Zeros

The Flipper Zeros will serve as versatile, portable hacking devices, capable of interacting with RFID systems, IoT devices, and wireless protocols like NFC and Wi-Fi. These multifunctional gadgets will allow us to simulate a variety of physical and wireless-based cyber-attacks in a real-world context. Their ability to interface with hardware like RFID tags, access control systems, and various IoT devices makes them crucial for exploring both the offensive and defensive aspects of cybersecurity. The Flipper Zeros will be used to test how well our network and security infrastructure hold up against different forms of wireless attacks and will help us assess how such attacks can be detected and mitigated by Zeek and our visualization system.

Breadboard and Wires

The breadboard and wires will connect the RFID scanner or other hardware-based security features to the Raspberry Pi server. This hardware is essential for prototyping any additional physical components (e.g., sensors or scanners) and integrating them into the network. It helps with experimenting with IoT security aspects, complementing the cyber-attack simulations in the network.

Cyber-attack Simulation Tools (e.g., Metasploit, Scapy)

These tools will be used to simulate cyber-attacks on our Raspberry Pi network, allowing us to test Zeek's detection capabilities and gather data for our visualization system. Simulating attacks is crucial for validating our project. Tools like Metasploit and Scapy provide a realistic environment to test various attack types and monitor how Zeek responds.

Zeek

Zeek is an open-source network monitoring tool that will be installed on one of our Raspberry Pi servers. It will analyze traffic, detect network anomalies, and log cyber-attacks or suspicious behavior. Zeek is crucial to our project's goal of detecting and visualizing cyber-attacks. It will generate the network data that forms the foundation of our project's attack detection and situational awareness system.

Linux Distribution to Run on Servers

A lightweight Linux distribution, such as Raspbian or Ubuntu, will run on our Raspberry Pi boards to serve as the operating system for setting up the network environment and Zeek. Linux is ideal for running servers due to its flexibility, security, and low resource consumption. It provides a stable platform for hosting Zeek and other necessary network services, allowing us to simulate realistic cyber network conditions.

Database

The database will store the logs and traffic data collected by Zeek, allowing us to manage and query large datasets of network activity and attack information. This will make it easier to visualize patterns and anomalies. Having a robust database system (e.g., MySQL or PostgreSQL) is essential for efficiently storing and analyzing network traffic data. This is especially important for building the visualization component of our project, as it allows for complex queries and meaningful insights to be drawn from the data.

Section D. Concept Generation

Design 1: Generalized Attack Simulation

For our initial design concept, we considered a more broad and flexible approach to attack simulation, without focusing on a specific target system. The idea was to create a platform that could be used to detect and analyze a variety of potential cybersecurity threats, rather than limiting ourselves to RFID-based attacks.

Some of the key elements we discussed in this initial concept include:

- RFID Chips: Exploring how to monitor a network for anomalous RFID chip activity, which could indicate an attack or unauthorized access attempt.
- Electrical Grid Protection: Identifying abnormal patterns in power distribution data that could signal a cyberattack or imminent system failure.
- Autonomous Vehicle Security: Detecting anomalies in data streams from vehicle sensors that could suggest spoofing attempts or other malicious activity.
- General Sensor Data Monitoring: Monitoring any type of sensor data for strange patterns that could imply an attack or system failure, rather than focusing on a specific application.

The advantages of this broad approach were the potential for greater flexibility and the ability to apply the platform to a wider range of real-world scenarios. However, the lack of a specific target made it challenging to define clear design requirements and validation procedures. Additionally, the broad scope risked diluting our focus and resources.

Design 2: RFID-Focused Attack Simulation

Using our initial concept, our second design focused specifically on RFID systems as the target for attack simulation and detection. This allowed us to narrow our scope and develop a more tailored solution.

In this design, we envisioned the two-Raspberry Pi architecture as the core of the system, with each Pi serving a specific role:

- Door Reader Pi: Responsible for interfacing with RFID readers and handling access control requests.
- RFID Server Pi: Functioning as the authentication server, processing RFID data and managing the access control database.

The Flipper Zero would be programmed to host attacks on the respective Pi, and the network activity, if deemed anomalous, would be pinged.

Additionally, the design incorporated a laptop that ran the Zeek network monitoring tool to detect anomalies.

We also considered incorporating an antenna and oscilloscope to capture and analyze the wireless RFID signals, but this was deemed beyond our budget and scope.

The advantages of this approach were the ability to create a more focused and realistic test environment for RFID-based attacks, as well as the opportunity to develop specialized detection and visualization capabilities.

Design 3: Expanded Three-Pi RFID Attack Simulation

Building on the RFID-focused approach of Design 2, this version maintained the three-Pi structure but expanded the attack capabilities:

- Door Reader Pi: Responsible for interfacing with RFID readers and handling access control requests.
- RFID Server Pi: Functioning as the authentication server, processing RFID data and managing the access control database. This Pi also hosted the Zeek network monitoring tool.
- Attack Pi: Simulating a broader range of attacks, including brute force, MITM, cloning, and denial-of-service (DoS), using the Proxmark3 and Flipper Zero devices.

The key differences from Design 2 were:

- Reduced redundancy by integrating the monitoring functionality into the RFID Server Pi, instead of using a separate laptop.
- Removal of the plan to incorporate an antenna and oscilloscope, due to budget and time constraints, and instead focusing on the core three-Pi architecture.
- Dedicated Attack Pi to automate and aid in the deployment and execution of attack scenarios.

This design offered a more streamlined and comprehensive platform for RFID security testing, with clearer separation of roles and expanded attack scenarios compared to the previous iteration.

By focusing on RFID-specific attacks and leveraging the three-Pi architecture, we were able to create a comprehensive and adaptable platform for testing the security of RFID systems. The finalized design provided a balance between technical depth and practical feasibility, making it a strong candidate for further development and implementation.

Section E. Concept Evaluation and Selection

Throughout our project we have developed three main concepts on how we can proceed with our project. One where we use our computers as servers and buy additional equipment as needed. One where we would use three pi's, a proxmox, and two flipper zeros. Finally, we would use the aforementioned setup but also acquire an antenna and oscilloscope to potentially increase the amount of types of attacks and ways to detect attacks. Each one of these concepts has its own advantages and disadvantages.

When using our personal computers we would have a few advantages mainly being immediate cost. Using our own computers would make it so we would not have to buy Pi's or another device to run a server or attacks through. However this also comes with the disadvantage of potentially needing to purchase additional equipment to be able to effectively acquire data and carry out certain tests. Additionally there could be issues with hardware limitations and compatibility of software that we would want to use. For example, Zeek was not made for Windows systems.

When using Pi's for the server we gain a lot of compatibility and customization with a bit more of an upfront cost. Pi's are rather inexpensive and are very simple making them great for a lot of applications and rather intuitive to use. We only need to buy a few things such as a proxmox for scanning and flipper zeros for attacking to get the ability to run tests with this approach. Setting up the software is where the tricky part of this approach is being that we need to setup the pi's from scratch.

Adding an antenna and oscilloscope to either of these could increase the amount of data we could extract from tests and allow us to run different types of tests giving us a wider scope to our project. The issue them comes to complexity and staying within scope and making sure we don't over exert ourselves with work. There is also an increase in cost because in addition to the new equipment we would need to buy extra hardware to be able to communicate with the antenna and scope.

	Pi's + Flipper + Proxmox	Pi's + Oscilloscope + Antenna + Flipper	Personal Computers + Flipper + Reader
Cost	2	1	2
Complexity	2	1	2
Implementation	2	2	1
Possible Attacks	2	3	2
Total	8	7	7

Note: Lower score is worse

Section F: Design Methodology

Our project focuses on setting up a network infrastructure to simulate real-world cyber-attacks using Radio Frequency Identification (RFID) signals and developing a system to detect and mitigate these attacks. This section outlines the methods employed to evaluate, refine, and validate the design through an iterative engineering process.

F.1 Computational Methods

We will utilize computational methods to analyze and validate the effectiveness of our attack detection algorithm. This includes:

- **Simulation of Attacks**: Using Python scripts on the Flipper Zeros, we will simulate brute force and Denial of Service (DOS) attacks on the Raspberry Pi authentication server. Computational models will evaluate the system's response under varying conditions, including increased signal frequency and combinations of legitimate and malicious signals.
- **Signal Filtering Algorithm**: Our algorithm will employ statistical methods, such as frequency analysis and anomaly detection, to classify incoming signals as legitimate or malicious. The filtering logic will be tested and refined based on the simulation results.

Software Tools:

- **Zeek Network Security Monitor**: Used for traffic analysis and detection of anomalies.
- **Kali OS**: Running on all three Raspberry Pi servers, Kali OS provides a comprehensive suite of network monitoring and penetration testing tools, facilitating simulation and analysis of attack scenarios.

Boundary conditions for these computational models include controlled network traffic, predefined signal parameters for legitimate communication, and attack variations generated by the Flipper Zeros.

F.2 Experimental Methods

Experimental testing will be conducted to validate the functionality of the network setup and attack detection system. Key steps include:

- **Setup Layout**: The experimental environment consists of:
  - **Authentication Server**: A Raspberry Pi configured to verify RFID signals and run the attack detection algorithm.
  - **Raspberry Pi Servers**: Two Raspberry Pi devices function as RFID servers, receiving signals and relaying them to the authentication server.

    - **Proxmark Devices**: One Proxmark is connected to each Raspberry Pi RFID server to scan and transmit RFID card signals.
    - **Flipper Zeros**: Two devices simulate attacks by generating brute force and DOS signals.
- **Testing Equipment**:
    - **RFID tags and readers**: To generate legitimate signals.
    - **Network analyzer**: To monitor traffic and server behavior during attacks.
- **Procedure**:
    - Run baseline tests to establish normal server behavior with legitimate RFID signals.
    - Introduce controlled brute force and DOS attacks using the Flipper Zeros.
    - Collect data on server response times, signal accuracy, and detection efficiency via Zeek.

F.3 Architecture/High-level Design

The high-level system architecture is designed to emulate a secure RFID-based authentication network while exposing it to vulnerabilities.

- **Signal Flow**:
    - RFID signals are sent to the Raspberry Pi servers for processing.
    - Signals are relayed to the authentication server for verification.
- **Attack Simulation**:
    - The Flipper Zeros are programmed to perform brute force and DOS attacks.
    - These attacks test the robustness of the network and detection algorithm.
- **Zeek Monitoring**:
    - Zeek scans the network for unusual patterns, such as excessive traffic from a single source or irregular timing intervals.

F.4 Validation Procedure

To validate that the final design meets the client's needs, the following steps will be taken:

- **Prototype Testing**:
    - Deploy the full system in a controlled environment to test attack detection and filtering accuracy.
    - Use performance metrics such as false positive/negative rates, response time under attack, and legitimate signal throughput to assess the algorithm.
- **Client Demonstration**:
    - Schedule a meeting to present the prototype.
    - Demonstrate system functionality, including normal operation, brute force attack mitigation, and DOS attack handling.

    - Provide live visualization of network traffic using Zeek.
- **Feedback Collection**:
    - Conduct a structured interview with the client to capture feedback on the system's performance and usability.
    - Distribute a formal survey for quantitative assessment of the project's success in meeting objectives.
- **Iterative Improvements**:
    - Incorporate feedback from the client and faculty advisor.
    - Conduct follow-up tests to ensure the design meets revised specifications.

Section G. Results and Design Details

G.1 Results and Design Details

Design Details

Our system design integrates multiple hardware and software components to establish a portable, scalable platform for RFID security research and monitoring. The current setup leverages **Raspberry Pi** for computing, **Proxmark3** for RFID interaction, and several supporting tools for log collection and analysis. While the installation of **Zeek** on the Raspberry Pi is still underway, its anticipated role in the design has been accounted for in future stages.

Hardware Integration:

- **Raspberry Pi**: The Raspberry Pi is central to the system, providing the computational backbone for processing RFID data and hosting the tools. Its portability and power efficiency make it ideal for field deployments. The Raspberry Pi also acts as a bridge between Proxmark3 and additional peripherals.
- **Proxmark3**: The Proxmark3 hardware handles all RFID-related operations, including reading, writing, cloning, and emulating tags. Its dual-frequency antenna setup enables communication with both low-frequency (125/134 kHz) and high-frequency (13.56 MHz) RFID tags. The Proxmark3 is interfaced with the Raspberry Pi via USB.
- **Peripheral Devices**: Additional devices, such as an NFC reader and external storage for logs, complement the core hardware, extending the system's capability for RFID data acquisition and storage.

Software and Tools:

- **Proxmark3 Client Software**: Installed on the Raspberry Pi, the Proxmark3 client provides an interactive interface for executing RFID operations. The client logs all interactions, such as tag reads, writes, and emulations, which can later be analyzed.
- **Custom Logging Scripts**: Python scripts are in development to automate log collection and basic event detection. These scripts are designed to capture Proxmark3 logs and prepare them for integration with Zeek once the installation is complete.
- **Zeek (Planned)**: Zeek will serve as the primary network and system monitoring tool, analyzing logs generated by the Proxmark3 client and identifying anomalous RFID activities. While Zeek is not yet operational, its expected role includes detecting replay attacks, unauthorized cloning attempts, and other malicious behaviors in RFID environments.

System Workflow:

1. **RFID Interaction**: Proxmark3 reads and processes data from RFID tags, storing log files locally on the Raspberry Pi.

2. **Log Forwarding**: Custom scripts forward these logs to a designated monitoring system. Once Zeek is operational, it will process these logs for advanced threat detection.
3. **Real-Time Alerts**: While advanced monitoring is pending, preliminary alerts are generated based on predefined conditions in the custom scripts (e.g., flagging repeated cloning attempts).
4. **Future Integration**: Upon successful Zeek installation, the system will include real-time monitoring, threat detection, and alerting features for RFID operations.

G.2 Results

Current Progress:

- The Raspberry Pi and Proxmark3 have been successfully integrated, allowing seamless RFID operations. Basic tasks like reading, cloning, and emulating tags have been tested with satisfactory results.
- Custom scripts for log collection are under development. These scripts currently capture logs from the Proxmark3 client and organize them for manual review and analysis.
- Preliminary testing demonstrated the Raspberry Pi's ability to handle Proxmark3 commands effectively, even under load.

Observations from Initial RFID Tests:

1. **Tag Operations**:
    - Successfully read low-frequency and high-frequency tags, including Mifare Classic and HID Prox tags.
    - Basic cloning and emulation were achieved using the Proxmark3 client, validating the system's capability to replicate RFID data accurately.
2. **Log Management**:
    - Logs generated by Proxmark3 are being stored on the Raspberry Pi and partially processed using basic filtering scripts.
    - Without Zeek, analysis is currently manual, but results indicate that tags can be cloned and replay attacks simulated effectively.

Challenges and Current Limitations:

- **Zeek Installation Pending**: The integration of Zeek is incomplete, limiting the system’s ability to perform real-time monitoring and anomaly detection. Current logs are stored locally without automated processing.
- **Data Overload**: Logs generated during high-frequency operations require substantial manual effort to sift through. This will be addressed once Zeek or an alternative log management tool is functional.

- **Performance Constraints**: The Raspberry Pi occasionally experiences lag during high-intensity RFID operations, indicating potential hardware limitations for concurrent logging and processing.

Preliminary Insights:

- The system has already demonstrated its potential to detect vulnerabilities in RFID systems, such as weak encryption on Mifare Classic tags and the feasibility of cloning HID Prox tags.
- The current logging framework, while basic, captures essential events that can be used to trace suspicious activities manually.

G.3 Next Steps

1. **Complete Zeek Installation**:
   - Finalize the setup and configuration of Zeek on the Raspberry Pi. This will allow for automated log processing, real-time monitoring, and anomaly detection.
   - Configure Zeek scripts to analyze Proxmark3 logs for patterns indicative of RFID attacks.
2. **Enhance Log Management**:
   - Implement a more sophisticated log forwarding and aggregation system, possibly integrating with cloud-based platforms like Splunk or Elastic Stack.
   - Develop visualization dashboards to simplify the analysis of RFID operations.
3. **Expand Testing**:
   - Conduct further testing with a broader range of RFID tags and systems to evaluate the system's versatility and effectiveness in diverse scenarios.
   - Test the Raspberry Pi's performance under concurrent RFID operations and monitoring tasks.
4. **Optimize System Performance**:
   - Explore hardware optimizations for the Raspberry Pi, such as overclocking or offloading tasks to external systems, to improve its processing capability.

This revised architecture demonstrates the system's foundational strength and highlights the next steps required to achieve full functionality. While the integration of Zeek remains pending, the current progress with Proxmark3 and basic logging scripts provides a solid base for further development. Once complete, the system will offer a powerful combination of RFID testing tools and real-time monitoring capabilities.

Section H: Societal Impacts of Design

H.1 Public Health, Safety, and Welfare

Our design focuses on simulating and detecting cybersecurity threats, which have significant implications for public health, safety, and welfare by advancing the security of RFID-based systems.

- **Design Safety Features**:
    - **Signal Filtering Algorithm**: Differentiates legitimate and malicious signals to ensure only authorized access is granted. This feature prevents potential security breaches that could compromise sensitive systems.
    - **Attack Detection via Zeek**: Monitors network traffic in real time, providing alerts for unusual activity, which enhances situational awareness and mitigates threats effectively.
    - **Isolated Network Environment**: The experimental setup uses a sandboxed environment, eliminating risks to external systems during testing.
- **Impact on Public Health, Safety, and Welfare**:
    - Enhanced cybersecurity reduces risks associated with unauthorized access to critical systems, such as medical devices or infrastructure.
    - Mitigating attacks like DOS ensures reliable access to services that the public depends on, thereby improving overall safety and welfare.

Codes and standards referenced include the **National Institute of Standards and Technology (NIST) Cybersecurity Framework**, which informed the design of secure authentication processes and attack mitigation techniques.

H.2 Societal Impacts

By improving the resilience of authentication systems, our design contributes to societal trust in technology and digital infrastructure.

- **Enhanced Security Awareness**: Demonstrating vulnerabilities and mitigation techniques can educate engineers, stakeholders, and users about potential risks, fostering proactive cybersecurity measures.
- **Impact on Daily Interactions**: Secure systems reduce the likelihood of data breaches, ensuring safer use of technologies like access control systems in workplaces, schools, and public facilities.

H.3 Political/Regulatory Impacts

Our project aligns with and informs regulatory frameworks surrounding cybersecurity:

- **Compliance with Standards**: Simulated attacks and detection systems can be used as benchmarks to validate compliance with regulatory standards, such as those outlined by **ISO/IEC 27001**.
- **Influence on Policy**: Demonstrating the vulnerabilities of RFID systems could encourage policymakers to strengthen guidelines for secure design in IoT and authentication systems.
- **Potential Risks**: The misuse of tools like the Flipper Zero for malicious purposes underscores the importance of robust regulations governing the sale and use of cybersecurity devices.

H.4 Economic Impacts

The design has both short- and long-term economic implications:

- **Cost Savings for Businesses**: Improved detection systems reduce the financial impact of cyber-attacks, which can cost millions in damages.
- **Market Potential**: The technology could influence the cybersecurity market by highlighting gaps in current solutions and prompting the development of new products.

H.5 Environmental Impacts

While the project uses energy-intensive devices, we have taken steps to minimize its environmental impact:

- **Energy Efficiency**: Raspberry Pi devices are low-power alternatives to traditional servers, reducing the carbon footprint of our setup.
- **Waste Minimization**: The equipment used, such as Proxmarks and Flipper Zeros, is durable and reusable for future educational or experimental purposes.
- **Potential Negative Impacts**: The production of electronic components has environmental costs, including e-waste and resource extraction.

H.6 Global Impacts

The project's focus on cybersecurity has global implications:

- **International Collaboration**: The techniques developed can be shared across borders to strengthen global defenses against RFID-based threats.
- **Global Risks**: The misuse of attack simulations by malicious actors could have far-reaching consequences if not carefully managed.

H.7 Ethical Considerations

Our project raises important ethical questions:

- **Responsible Use of Technology**: Demonstrating attack techniques like brute force and DOS requires safeguards to ensure that these methods are not misused.
- **Privacy Concerns**: The design must ensure that testing does not inadvertently collect or expose sensitive data.

By carefully considering these factors, we aim to create a design that not only meets technical specifications but also contributes positively to society while mitigating any negative or unintended consequences.

Section I. Cost Analysis

All purchases have been made through VCU's Office of Procurement Services, with a total budget allocation of $1,000 USD. Current expenditures are tracked below, showing we have used $806.78 of our allocated budget.

Date Ordered	Date Received	Item Description	Quantity	Unit Price	Total Price	Vendor	Reference Number
10/22/2024	10/23/2024	Proxmark3 Easy V3.0 RDV4	2	$76.00	$152.00	Amazon	P0252162
10/22/2024	10/23/2024	Raspberry Pi 5 Case	3	$14.84	$44.52	Amazon	P0252162
10/22/2024	10/25/2024	Raspberry Pi 5 (2GB RAM)	3	$58.79	$176.37	Amazon	P0252162
10/24/2024	10/29/2024	Flipper Zero	2	$169.00	$338.00	Flipper Devices, Inc.	#490227
10/30/2024	10/31/2024	32GB microSD Cards	3	$11.39	$34.17	Amazon	P0253351
11/06/2024	11/13/2024	Power Supply for RPi 5	3	$12.98	$38.94	Amazon	P0254139
11/08/2024	11/13/2024	32GB microSD Cards	2	$11.39	$22.78	Amazon	P0254591

Budget Summary

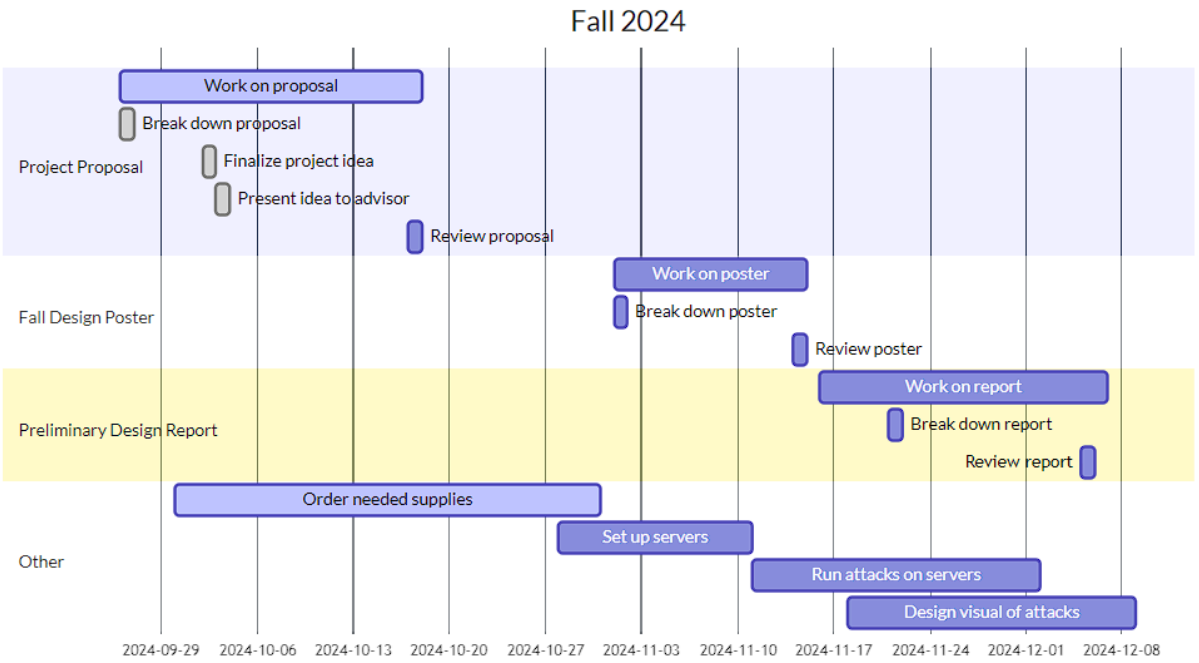
- Total Budget Allocated: $1,000.00
- Total Spent: $806.78
- Remaining Budget: $193.22
- Budget Utilization: 80.68%

Section J. Conclusions and Recommendations

We spent the first few weeks researching how RFID works, the types of attacks there are, and what processes we can take to detect them. We had meeting every week to talk about our findings and what to work on next. We had a few ideas on what designs we could pursue with ideas of having antennas and trying to detect small anomalies in the frequencies submitted from attackers but we eventually decided on running our attacks through three raspberry pi's a proxmox and two flipper zero's. The pi's would be split into one for requesting, one for rfid. and one for authentication. The proxmox would be used to capture rfid signals and the flipper zeros would be our attackers in our tests. We are currently working on the final steps to make it so we can start running tests. We are having trouble getting Zeek to run on correctly on the Pi's so over break we hope to fix the issue.

As a result of not being able to run any tests we do not have a final product to know where it can be improved or what to improve however, some potential improvements that could be made are the addition of antennas and a oscilloscope to increase the amount of detectable attacks. This is probably too expensive for us to acquire without additional funding and would have to be reserved for the next years capstone team if they wish to pursue our design.

Appendix 1: Project Timeline



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

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 brings to the group*	*Other Info*	*Contact Info*
Jaspreet Singh	Cyber Security Experience and a handful of knowledge	I enjoy being a part of the Cyber security team.	Singhj21@vcu.edu
Christopher Malatesta	Jave experience and a little bit of C	Hoping to learn about CyberSec as we progress through the project	malatestacn@vcu.edu
Keith Tran	Cloud Experience	Not a cyber focus, software, but can learn quick	trand15@vcu.edu
Calvin Hurlbert	Java, C, Operating Systems, and Algorithms	I don't have any cybersecurity experience but I'm very excited to learn.	hurlbertc@vcu.edu

Other Stakeholders	*Notes*	*Contact Info*
Milos Manic	Faculty Advisor	mmanic@vcu.edu
Bjorn Vaagensmth	Project Sponsor	bjorn.vaagensmith@inl.gov

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 cultural goals? These are your Actions (middle column). Finally, how do students deviate from the team's cultural 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 of Team Culture can be found on the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

Culture Goals	***Actions***	***Warning Signs***
Being on time for every meeting	- Set up meetings in Discord - Send a reminder Discord message in day before the meeting.	- Student misses first meeting, warning is granted. - Student misses meetings afterward – the issue is brought up with the faculty advisor
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

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, 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	Every Thursday at 06:00 PM on Discord	Update the group on day-to-day challenges and accomplishments. Completing anything that needs to be done as a group.
Students Only	As needed	Actively work on projects. Finalizing deliverables if needed.
Students + Faculty advisor	Every Thursday at 10:00 AM through Zoom	Update the faculty advisor and get answers to our questions.
Project Sponsor	Every Thursday at 10:00 AM through Zoom	Update the 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 the 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 about 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 the 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 an overall schedule for the 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; leads in establishing contact within and outside of the 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 the 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 findings and resulting recommendations.
6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

Team Member	***Role(s)***	***Responsibilities***
Chris Malatesta	Project Manager	Manages the project as a whole and keeps a schedule for the project. Runs meetings and ensures everybody completes their tasks.
Keith Tran	Financial Advisor	In charge of budgeting for the group and keeping track of purchases. Carries out purchases on behalf of the team after discussion with the group.
Calvin Hurlbert	Logistics Manager	Tasked with staying up to date with all communication with the faculty advisor and the project sponsor. Keeps the team updated with all communication and reviews during team meetings.
Jaspreet Singh	Systems Engineer	Summarizes the client’s design specification and breaks down tasks for everyone to complete. Make sure the GitHub is up to date and all branches are merged properly.

Step 5: Agree to the above team contract

Team Member: *Signature: Christopher Malatesta*

Team Member: *Signature: Jaspreet Singh*

Team Member: *Signature: Keith Tran*

Team Member: *Signature: Calvin Hurlbert*