Project Name
Vision and Scope

Stephen Loudiana, Kevin McGrane, Dylan Roy

November 18, 2021

Version: 1.0

Ver.	Date	Who	Change	Who

1 Business Requirements

1.1 Background

Currently, in NetBSD, in order to connect to Wi-Fi the user must manually connect, utlizing wpa_supplicant and IOCTL to determine the Wi-Fi information, constructing a configuration file, and then starting wpa_supplicant with the configuration file to connect. In contrast, Windows users connect to Wi-Fi using an API through a GUI, in which the user does not need to type specific commands into the command line interface, but rather click a few buttons, enter a password, and they're finished (generally).

1.2 Business Opportunity

There is an opportunity to bring similar functionality that Windows currently has in connecting to wifi networks to NetBSD, ultimately streamlining the process in which a NetBSD user connects to wifi networks. As an affect of this addition, time and effort spent in connecting to a wifi network will be greatly reduced, thus potentially providing a more inviting NetBSD experience.

1.3 Business Objectives and Success Criteria

The customer needs this product to help minimize the time and effort spent connecting to Wi-Fi in NetBSD. The customer will know the product is a success if the user is able to effortlessly connect to Wi-Fi in less time. Creating an API to streamline the process would take less effort for the typical user. We can determine if our product is a success by thoroughly testing it before deployment and verifying that wifi connecting is indeed a simpler process than it originally was.

1.4 Customer and Market Needs

Meeting the objectives successfully would enhance many NetBSD users' experience within the operating system, especially those who spend much time connecting to wifi networks. The NetBSD has had a request for this functionality for many years, and thus it was fitting for our customer to source a solution through this project which would directly address the request.

1.5 Business Risks

We can determine our biggest failure if our API is not efficient in connecting to Wi-Fi, or if the API does not work altogether. However, business risks and hazard is very low. We will thoroughly test our product before deployment to ensure it works before customers use the new system. If our API does not work, customers will still be able to connect to Wi-Fi using the wpa_client. This puts our risk very low as our API will only work to improve a system where no current solution exists. We can work to protect against this by thoroughly testing our product before deployment.

2 Vision of the Solution

The solution will look like an API to interact with WPA Suplicant and IOCTL and a user interface to interat and use the API to connect to wifi

2.1 Vision Statement

For: NetBSD users
Who: Need to connect to wifi in a timely manner
The: NetBSD Wifi Browser
Is: A user interface
Unlike: Any other solution within NetBSD
Our Product: Will streamline the connection to wifi

2.2 Major Features

The major features that address the customer's needs is the API for handling the communication with wpa_supplicant and IOCTL, removing the necessity for a user to work through those mediums. To further simplify the process, multiple user interfaces will be provided with varying levels of simplicity to allow a user maximum flexibility with connecting to a wifi network.

2.3 Assumptions and Dependencies

Our product is dependent on the wpa_supplicatn and IOCTL programs not changing significantly during the development of our product as our API will direclty interface with the existing functionality of both programs. [Need to fill some assumptions maybe]

3 Scope and Limitations

Within the scope of the next 8 months in which we have to work, we will accomplish three major tasks: the creation of an API which interfaces with wpa_supplicant and IOCTL, a command line interface built on top of the API, and a terminal user interface also built on the API, both of which will allow a user to easily interact with wpa_supplicant to connect to a wifi network. If time and development progress is permitting, a GUI in the style of one of the popular desktop environments of NetBSD will also be created similarly to the CLI and the TUI. An aspect that will have to wait for later projects and probably other teams is the development of GUI interfaces for each of the popular desktop environments. Out of scope for the wifi browser project is the porting of the applications and API to other UNIX or BSD based operating systems.

3.1 Scope of Initial Release

Our initial release of the product will certainly feature the API and the command line interface. This will meet the bare minimum of the product requirements and provide a foundational perspective of the future of the product, as well as inform the changes that may be needed from either our API or CLI.

3.2 Scope of Subsequent Releases

The release immediate proceeding from the initial release will be the creation of a terminal-base user interface. Subsequent releases will most likely focus on the implementation of proper GUI's built on the API, and, in the event that wpa_supplicant or IOCTL change significantly, updating our API to continue proper interfacing with both programs.

3.3 Limitations and Exclusions

Our major limitations for this project is the team having minimal experience with the NetBSD operating system and the amount of time we will have to work towards our goal. On top of this limitation, we will need to allot time to understand how to use wpa_supplicant and IOCTL to connect to wifi networks of all standards and protocols so that we properly design an API to do that for us.

4 Business Context

4.1 Stakeholder Profiles

The stakeholders for this project would be our client, the development team, the eventual users, and the future maintainers. The current development team in this instance would be us, the eventual users would be anyone that is currently or who will use NetBSD in the future, and the future maintainers are the NetBSD development team who will eventually become maintainers if the system is passed to them. Those who will benefit from our project include users of NetBSD.

4.2 Project Priorities

We will consider developers to be our greatest stakeholders, and then customers or investors of our project. However, we see very little return on investment for our project. NetBSD is a free open source operating system, so our API will be free for the NetBSD community.

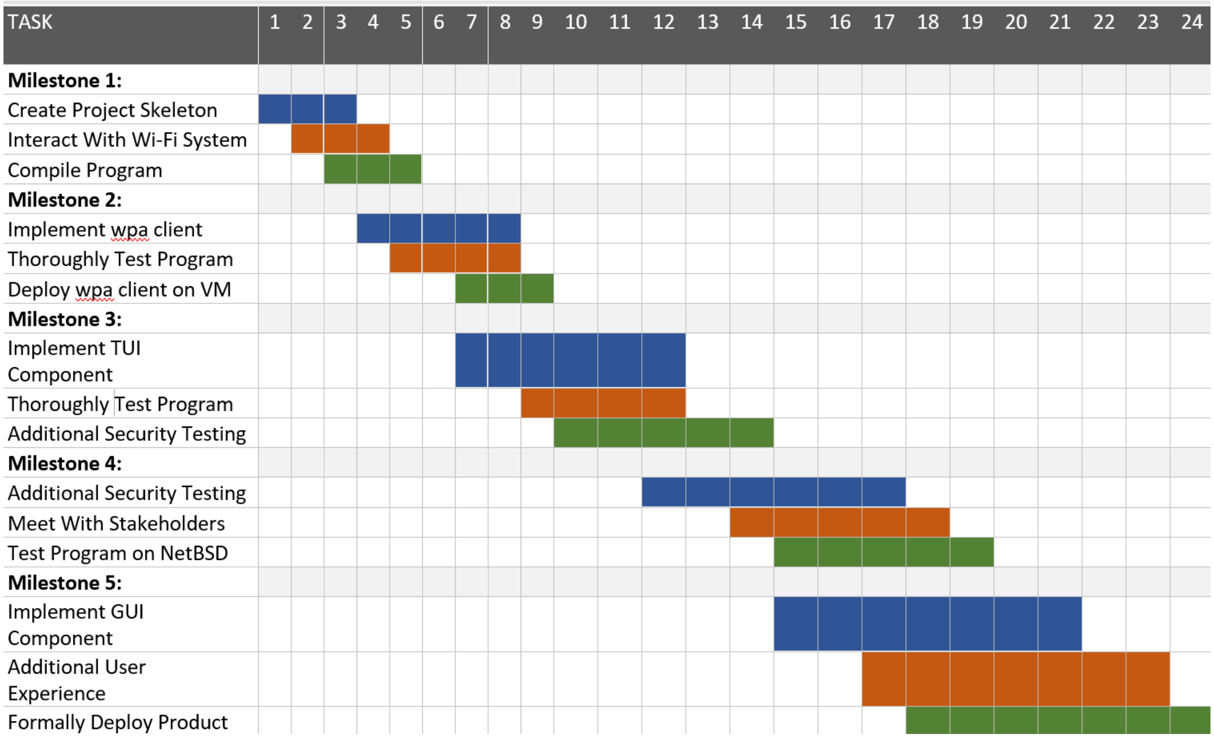
4.3 Operating Environment

The product will be used as an interface to easily connect to Wi-Fi when using NetBSD. It will only be used by the NetBSD operating system.

5 Schedule

5.1 Gantt Chart

We went with the agile method here. Time indications show number of weeks.



TASK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Milestone 1:																								
Create Project Skeleton	🟦	🟦	🟦																					
Interact With Wi-Fi System		🟧	🟧	🟧																				
Compile Program			🟩	🟩	🟩																			
Milestone 2:																								
Implement wpa client				🟦	🟦	🟦	🟦	🟦																
Thoroughly Test Program					🟧	🟧	🟧	🟧																
Deploy wpa client on VM							🟩	🟩	🟩															
Milestone 3:																								
Implement TUI Component							🟦	🟦	🟦	🟦	🟦	🟦												
Thoroughly Test Program									🟧	🟧	🟧	🟧												
Additional Security Testing										🟩	🟩	🟩	🟩	🟩										
Milestone 4:																								
Additional Security Testing												🟦	🟦	🟦	🟦	🟦	🟦							
Meet With Stakeholders														🟧	🟧	🟧	🟧	🟧						
Test Program on NetBSD															🟩	🟩	🟩	🟩	🟩					
Milestone 5:																								
Implement GUI Component															🟦	🟦	🟦	🟦	🟦	🟦	🟦			
Additional User Experience																	🟧	🟧	🟧	🟧	🟧	🟧	🟧	
Formally Deploy Product																		🟩	🟩	🟩	🟩	🟩	🟩	🟩

5.2 Key Milestones

Milestone 1: Create skeleton of project. Skeleton will include any methods we think will help in creating our project. Will also contain header files that will link all files together, as well as a makefile to make an executable of our project. For our skeleton to be successful, we will need to make sure we are interacting with the system of NetBSD that is responsible for connecting to Wi-Fi, as well as streamlining the process through wpa_client. For milestone one, we are only interested in setting up the proper procedures to interact with NetBSD and wpa_client with minimal functionality.

Milestone 2: Ensure product is working correctly. With our implementation and skeleton files ready, we can start to create the proper methods and procedures needed to make sure our system is secure. Set up additional methods to connect to Wi-Fi through wpa_client.

Milestone 3: Thoroughly test project to make sure it is working properly. Begin development on TUI that will interact with our system. Focus on software security as well as efficiency during milestone 3.

Milestone 4: Begin development of GUI to interact with our system. Meet with stakeholders as needed. Thoroughly test product on virtual machine to ensure it works correctly and streamlines the process of connecting to Wi-Fi.

Milestone 5: Meet with additional stakeholders at the time of deployment. Look for vulnerabilities in software and how to mitigate each vulnerability. Clean up any additional code and ensure product is working correctly.

5.3 Resource Assignments

Resources will include Western computers that can interact with NetBSD operating system so we can test our product. Additional resources will include additional computers to work on developing our project. Budgets and additional consumables I don't think are needed here.

5.4 Individual Responsibilities

Stephen Loudiana - Responsible for completing work not done on campus (stakeholders / writing) Kevin / Dylan - Split responsibilities for working on campus (interacting with vm, etc.)

6 Deliverables

- API
  The API would be the central component of the product produced, serving as the true mediator between the user and communicating with wpa_supplicant and IOCTL to accomplish the task of connecting to a wifi network.
- Command Line Interface (CLI)
  The CLI would allow the user to connect to a wifi network easiliy utilizing the command line and the aforementioned API.
- Terminal User Interface (TUI)
  The TUI would a curses based interface that would enable a user to connect to wifi more visually, presenting information without the necessity of a series of commands, but rather through automatically presenting the necesasry imformation

6.1 Documentation

The main user documentation that will come with our deliverables will be man pages to explain how to use the system.

6.2 Key Presentations

At the end of CSCI 492, ideally we will have an API to present to our peers with a semi-functional CLI to show off its capabilities and features. By the end of CSCI 493, at minimum we will have a complete CLI and TUI to present, and if development allows, a GUI (themed by one of the popular NetBSD desktop envrionments) to show to final functionality of our API and the success of simplifying the wifi connection process.