Milestone #2

NetBSD Wifi Browser Project 2022

Dylan Roy, Stephen Loudiana, Kevin McGrane

1 General Progress Report

Progress has been significant for the duration of Milestone 2, but not necessarily in the ways outlined in the development plan. Instead of all members focusing on the API to start out, we split the work into the three main components our projects encompasses: Dylan on the API, Stephen on the CLI, and Kevin on the TUI. The API's progress had many road blocks in its development due to the numerous undocumented quirks of interfacing with `wpa_supplicant`, though, as the roadblocks are overcome, progress on the API should speed up as the implementation beyond the quirks is rather trivial. In the development plan document, the next milestone might look a bit more ambitious, but the majority of the API implementation planned is closely related to each other and *should* have trivial implementation once the configuration file saving issue is solved. Making up for the unexpected delays in the API, Stephen has put a significant amount of work into the CLI, saving us time that was allocated for later in the quarter originally, negating (hopefully) the API slow downs.

The progress on the TUI has been slow to get started due to the fact that it has mostly been filled with research and comparing with other TUIs to see the best plan of action when trying to implement it. It seems that Ncurses is going to be our best bet when trying to efficiently implement a TUI that works with the API, thus a large majority of the our time during milestone has been used on researching Ncurses and how to use it when implementing the TUI. There was also some confusion at the start of our implementation on how the TUI will interact and work with the API, but after a few group discussion those gray areas were cleared up and have really set us up for success moving into the future.

Accomplishments

- API can communicate with a running instance of `wpa_supplicant`
- Projects can use the API in its current state and use some functionality

Targets Not Met:

- API has not independently facilitated `wpa_supplicant` connecting to a wifi source of any kind
- API is not hashing passwords yet
- API can remove networks from configuration, but not externally yet
- API testing is not really a suite yet, but still being built out

2 Git Branch Report

All work has been performed on `master` against best practice. It might actually behoove us to work on separate branches due to the experienced difficulties with people working on master at the same time.

3 Individual Progress

Kevin: ¡time¿

- Research on best way to implement TUI by comparing with other examples and working TUI systems
- Research on Ncurses and how it will be implemented
- Began some sample testing with Ncurses
- Review of API .h file and how it is going to interact with the TUI
- Research and problem solving on already existing issues with Ncurses when implementing a TUI

Dylan: ¡time¿

- Set up remote connection to lab machine
- Built API scaffolding and function prototypes
- Extensive additional into interfacing with `wpa_supplicant`
- Linked `wpa_ctrl` - the `wpa_supplicant` interface - to the API
- API can compile with `wpa_ctrl` files and a program file
- Connected the API to `wpa_supplicant`
- API can list available networks in proximity to the wifi interface
- API can list configured networks in the configuration file `wpa_supplicant` is currently using
- API can temporarily create new network configurations and write to them, but they don't save when `wpa_supplicant` is asked to re-read the configuration file
- API can create a local configuration file, but `wpa_supplicant` terminate when trying to use it mysteriously
- Building out a document of interfacing with `wpa_supplicant`