Project:

Sentinel - Tiered Account Manager (C & Haskell)

Description:

For awhile now, I have had an idea for an account manager that generates and organizes passkeys for all kinds of accounts (not just web logins). The general structure would be as follows: all passkeys are members of a group and all groups are assigned a tier. A tier represents a level of security and describes an algorithm for key generation and a duration before expiration. All tier, group, and account information should, of course, be stored securely.

I have had the opportunity to implement this using Python and the Web2Py framework; however, I have always wanted to implement it in a lower-level language like C. This is simply because at higher levels of abstraction, more layers of code are employed between the application and the machine. This leaves more opportunity for bugs and more possible entry points for potential attackers. In addition, I want to implement a core that can be tested and used before it is deployed as a web service. After the introduction to functional programming we were given at the beginning of the quarter, I knew that such a security-based application should be written in a mathematically analyzable & verifiable language like Haskell. Implementing Sentinel in both languages will likely help me become more comfortable writing in Haskell as I will be able to see how the necessary components I already know how to build imperatively are translated into Haskell's paradigm.

I feel that the entire structure of the program may be a little ambitious for the allotted time, so I will aim to construct reliable text storage and retrieval in an encrypted manner by the end of the quarter.

Team:

David Tucker

Time Budget:

4 weeks

Timeline:

2013-02-15: generic file-based storage (C)
2013-02-22: generic file-based storage (Haskell)
2013-03-01: AES Encryption (C)
2013-03-08: AES Encryption (Haskell)