Lukasz Baldyga - c3538194 - l.baldyga7726@student.leedsbeckett.ac.uk

Lukasz's Password Manager

Introduction

A password manager might seem like a simple application to create, but in reality there's a lot more that goes on in the background than it seems, especially in my password manager. In this proposal, I will write what my password manager should be capable of by the end of development.

Goals

The main goal of this password manager is to be as secure as possible but also as open and free as possible, with many secure ways of recovery without centralization.

"Open" means that the source-code for this project will be licensed under the MIT License and the source-code would be free and open source (after the initial project would be finished). Anyone would be able to implement my work into theirs, in case my fork of the source woulds stop.

"Free" meaning as in libre or free-speech. This also means that the application must be as cross-platform as possible.

The security would need to also be top-notch, this means:

1. Encrypting everything.
2. Leaving as little meta-data as possible.
3. Making direct and shortest (meaning least hops) connections wherever possible
4. Using algorithms that are quantum resistant (meaning they can't be broken using a quantum computer) on top of other traditional algorithms
5. Making sure that even encrypted data isn't sent were it shouldn't be.
6. Using key stretchering to protect against GPU based attacks.
7. Built-in standalone TOR support, with built in bridge support enabled by default.

Please keep in mind that this is not an extensive list of security protections and methods and they may change depending on the development of the app, but it's safe to say at least 80% of the features above will make it into the final product.

Ideally, I would like my product to be used by journalists, human right's activists, whistle blowers and/or people in non-democratic countries, in countries where human rights are not recognized or in countries with censorship. These people usually require topmost privacy and secrecy in their lives.

Brief overview of main components

Front-end App

The front end application should be fairly simple and will allow users to:

1. Insert password details
    a) Website name
    b) Website URL
    c) Password
    d) Other optional fields
2. Sync their data with their other devices action
3. Account overview
4. See all of their passwords
5. Recovery functions:
    a) Offline secret sharing
    b) Paper recovery (example, example)
    c) Shared secret recovery (How to Share a Secret by Adi Shamir)
6. Settings
7. Send files to other devices (maybe)

Back-end App

The back end of the app will allow:

1. Offline storage
2. Passive sync with other devices
3. TOR connection, with bridges enabled by default.
4. Entropy generation
5. Password generation
6. Send and receive files (maybe)
7. Sync (different methods depending on network status, tried in order):
    a) Offline, locally.
    b) Online, with the help of the server to locate the address of the other device.
    c) If one or more devices is firewall-ed, create a special pass-through via clearnet
    d) If server is unavailable over clearnet, try over TOR

Server

1. Entropy generation service (Something like ones used by Cloudflare) in case the device is slow and/or doesn't produce enough of entropy fast enough.
2. TOR middle-point. Allows devices to connect together over TOR.
3. Points to IPs. Allows the devices to connect to each other over UDP without directly going through the server.
4. Clearnet middle-point. Allows devices to connect together through the server.