COMP 3004B - Deliverable 1

Team Name: Hyggelig Security
App Name: hyggelig secure
Team Members:

- Connor Stewart, 101041125
- Gabriel Valachi, 101068875
- Kavan Salehi, 101046945
- Kevin Sullivan, 100896774

What is your project?

hyggelig secure is an application to help users make use of and learn about basic security practices.

"Hyggelig" is the adjective of "hygge"(hyoo-gah), a Danish word used to convey a feeling of coziness, everyday happiness, almost like the spirit of a hug. In the UK and North America, hygge has become a lifestyle phenomenon where practitioners make an effort to do small pleasurable things like wearing warm socks, or drinking hot tea while wrapped up in a blanket.

With hyggelig secure, we want to apply that hygge ideology to practising basic security hygiene. Much like hygge, keeping secure while interacting with cyberspace can be a matter of little steps that add up to so much more. The app will provide short, easy-to-do, satisfying activities that users can do all in one place. Users can also learn about each security practice and why the practice should matter to them.

Why is it interesting?

- Users are critical to the success or failure of a cyber crime (e.g. phishing)
    - Educating users on cyber security topics can play a significant role in defending against cyber criminals
    - Many breaches happen due to weak/reused passwords.
    - When breaches occur, data is often unencrypted, allowing an attacker to leak secrets and cause damage with their gained access.
- Mobile phones are integrated into our lives and are heavily in use, e.g. on long trips or to quickly access banking information without a desktop computer nearby.
    - Convenience/quick access is causing users to implement weak passwords; for example by having all their credentials remembered in their web browser.
- While there are easy-to-use security-related apps on the market, many of those apps are behind a paywall, and most of the apps do not focus on educating the user.
    - Free (ie. unpaid) apps are generally collecting data on the user, then selling that data to other parties.
- Making good security/privacy practices less tedious and more convenient to users will help users adopt better security routines.

Why does the project make sense in a mobile form factor?

- People share sensitive information through many different email and messaging apps that are accessed through their mobile phones.
- People are increasingly accessing the Internet through a mobile device. Users need help creating and storing account passwords that are not easily cracked by potential attackers.
- Users keep a lot of personal data on their mobile devices and many users do not know how to keep that data safe, or do not make the effort to keep that data safe.

Functional Properties

1. Password Generator
    1.1. User can use the Password Generator to generate a password that cannot be easily guessed and cannot be feasibly brute-forced.
    1.2. User should be able to store a generated password in the Password Manager.
    1.3. User should be able to control the potential complexity - length, alphabet, additional characters - of generated passwords.
2. Password Manager
    2.1. Password database should always be encrypted when not running, or when active and locked.
    2.2. For forensic safety, user can only look at one password at a time.
    2.3. User should be able to manage passwords by tag.
    2.4. User should be able to back up their database of passwords locally.
3. Encryption Tools
    3.1. User should be able to generate a PGP public/private keypair.
    3.2. User should be able to store photos/personal information in an encrypted folder.
    3.3. User should have the option to also permanently delete the old unencrypted files.
    3.4. User should be able to encrypt/decrypt files either with a symmetric encryption key (ie. a password), or a public key (which also supports signing/verification).
    3.5. User can take the output from other applications and encrypt it directly, such that it's never unencrypted on disk
4. Bitcoin Wallet
    4.1. User should be able to generate a public and private address to send and receive Bitcoin, respectively.
    4.2. User should be able to view their balance on the public Bitcoin ledger.
5. Cybersecurity Help
    5.1. Users should be able to access a rationale of how the activity contributes to their security
    5.2. Users will be presented with a tutorial when using the app for the first time.
6. Maintainability and Backups
    6.1. The system must be capable of storing password backups in a cloud environment to prevent data loss if the phone is lost, destroyed, or stolen.

User Scenarios

Scenario Name: Ransomware Scare
Participating Actor Instances: Miles - Office Worker
Flow of Events:

1. Miles works at an office that was recently infected by ransomware. Miles now wants to make a new password for his Sympatico email account.
2. Miles opens hyggelig secure and taps on the “Passwords” button.
3. The application changes to a screen with the “Make a new password” and “Manage my passwords” buttons, and Miles taps on the “Make a new password” button.
4. hyggelig secure changes to a screen with options on password complexity (length, with a symbol, with a capital, with a number) [FP1.3]
5. Miles chooses a 10-character-long password with a symbol and a number, then taps the “generate” button [FP1.1]
6. The application shows Miles the generated password, and prompts him to save the password in the password manager

7. Miles gives the password a tag, “sympatico” [FP2.3], and saves the new password in the Password Manager [FP1.2]

Scenario Name: Business Accounts

Participating Actor Instances: John - Business Owner

Flow of Events:

1. John owns and runs a small business which manages the taxes and payroll information of various professionals and entrepreneurs. He is often on the move, so he primarily uses his mobile phone for communications. Due to growing identity theft, John would like an extra layer of security for transferring his clients’ sensitive information.
2. John emails his clients and requests them to create public/private PGP keys for themselves, then to send him their generated public keys [FP3.1].
3. Once John receives his clients’ public keys, he opens hyggelig secure, taps the “Encryption Tools” option, selects “Import Keys” and imports his clients’ public keys, after which he is returned to the Encryption Tools menu [FP3.4].
4. John selects the “Encrypt File(s)” feature. He selects each tax/payroll document and assigns the public key(s) of the intended recipient(s) to each document. Once finished, John selects “Encrypt”; all the files are encrypted in bulk and saved locally [FP3.2].
5. John sends each encrypted document to the respective client.

Non Functional Properties

1. Secure
    a. Data stored on disk by the system must be encrypted.
    b. Brute force (ie. guessing the password) must be the only potentially feasible method an attacker can access files encrypted by the system.
    c. Sensitive data, such as cleartext passwords, must only be present in memory when absolutely necessary; when no longer needed, they must be scrubbed immediately.
2. Ease of Use
    a. Application first-time setup should take less than 1 minute.
    b. Users should be able to learn how to use any individual feature within 5 minutes.
    c. Users should be able to complete the tutorial within 5 minutes.
3. Portability
    a. The Encryption Tools must be sufficiently compliant with OpenPGP (RFC 4880) so that users of the application and recipients of encrypted files can use common tools (e.g. GnuPG) to decrypt and verify their files.
4. Recoverability and Data Management
    a. Data must be easily recovered in the event the phone is lost, damaged, or stolen.
    b. Data must be easy to organize and manage both within the applications user interface, and inside the computer's file system
    c. Data must be easy to import and export between various devices
5. Reliability
    a. Information must be stored in a stable file format
    b. Application should run smoothly without interruption or failure
    c. Encrypted data and password archives should be resistant to file system corruption in the event the phone crashes midway through reading or writing to the file.

These non functional properties are required as the system stores the user’s passwords, and is intended for use by novice and experienced users.

Low-Fidelity Mockups

Figure 1: Hyggelig Home Screen

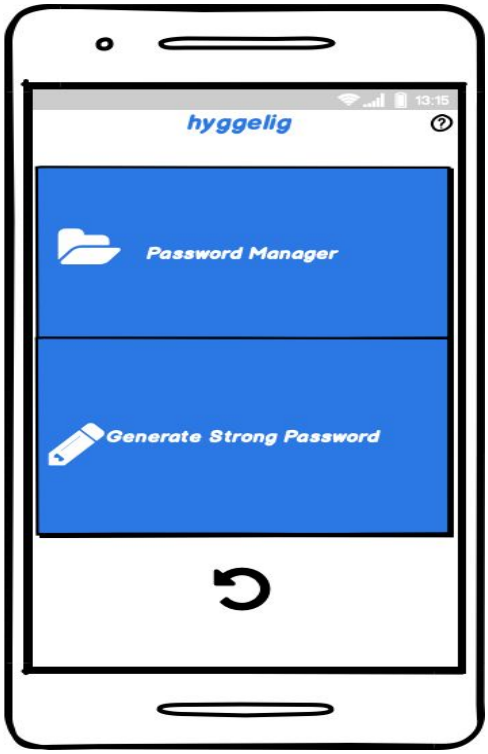

Figure 2: Hyggelig Password Tools Menu