Security and risk mitigation

This document serves as a small risk register for the project. It should be regularly reviewed.

Last review: 2022-08-26

Summary

Issue	Risk summary	Mitigation summary
Stored credentials	If an attacker got hold of the user's credentials, they could sign into the portal and conduct any malicious activity with the user's identity.	The app does not store user credentials, just a token with limited scope.
App secrets	If an attacker accesses the API directly, they could take actions or DDoS the portal.	Use of the `API-Key` header makes casual attacks harder. User token required to take actions. API endpoints are rate limited.
Data at rest	An attacker might be able to learn about the user from the app's logs or obtain the user's token to receive their notifications.	The app's database is encrypted at rest. Neither app logs or notifications contain PII.
Code	The code should be open sourced for public scrutiny, to develop trust.	The code should be reviewed to ensure it does not contain sensitive information (ie. keys, passwords, etc.)

Detail

Stored credentials

> If an attacker got hold of the user's credentials, they could sign into the portal and conduct any malicious activity with the user's identity.

The app does not store the user's credentials. Instead, it exchanges them for a user token, to be used on future communication with the portal.

As the device's push token may change at any time (at the whim of FCM or APNS), the user token is stored and used to resubmit the token if it refreshes.

Risks:

- The user token may be stolen.

Mitigations:

- The token is limited in scope - and cannot be used for general access to the user's account.
- The token can be cancelled if found to be compromised.

App secrets

> If an attacker accesses the API directly, they could take actions or DDoS the portal.

The app communicates with the portal using an API. The API requires a key, called `API-Key` to be provided with each call - and without it, the API will not respond. This key remains constant over time, and will be the same key used by every instance of the app.

The API key should not be considered a security feature, or a well-kept secret.

Strings in apps are vulnerable to reverse engineering and decompilation. Obfuscation offers some hope of making the API-Key *more difficult* to obtain, however it should not be considered a guarantee.

Mitigations:

- All user actions also require a user token (obtained by exchange of username and password).
- Server-side portal API endpoints are rate limited.

Data at rest

> An attacker might be able to learn about the user from the app's logs or obtain the user's token to receive their notifications.

The app employs a SQLite database to manage a small amount of data including push notification and portal configuration. This database is encrypted at rest, and this is configured through `SQLiteOpenFlags` in `DbConstants.cs`.

There are various options for this configuration:

- `ProtectionComplete` The file is encrypted and inaccessible while the device is locked.
- `ProtectionCompleteUnlessOpen` The file is encrypted until it's opened but is then accessible even if the user locks the device.
- `ProtectionCompleteUntilFirstUserAuthentication` The file is encrypted until after the user has booted and unlocked the device.
- `ProtectionNone` The database file isn't encrypted.

At current time, the app is configured for the `ProtectionCompleteUnlessOpen` option: The database is unlocked as needed, and then remains accessible to the app even if the user locks their device. This means that notifications received when the device is locked can still be recorded.

Resource: Xamarin.Forms Local Databases

Further, the content of the logs is carefully managed. PII is not recorded in the app's logs in the `RELEASE` build of the app (ie. the version that's available through the public app stores), nor is it included in the notifications sent to users. An attacker would need to know the user's username and password to be able to view the content of messages or events - and these credentials are not recorded by the app.

Code

This app should be made safe to open-source and share with the civic tech community.

Values for `API-Key` and credentials for test users are stored in a class called `SensitiveConstants` - which is not included in the repository.