Two-Factor Authentication for End-to-End Login

Systems Using Facebook Messenger

Privacy has become increasingly important in an increasingly technological world. Indeed, as more people are going online, we have also seen an increase in the amount of malicious attacks and hacks on unsuspecting users. There have been many ways to combat this. Many companies with login systems have required their users to create passwords that contain a certain number of characters. More recently, the SHA-1 cryptographic hash is becoming deprecated in 2017 in order to migrate to a more secure signing algorithm known as SHA-2, and websites are increasingly using the HTTPS protocol as opposed to the less secure HTTP protocol.

One way that login systems have attempted to increase security is by using two-factor authentication. Two-factor authentication is a method of access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism. At Yale, we use Two-Factor Authentication with our central authentication service (CAS) login. Users who login on CAS and are not on a Yale WiFi network must verify their login using a mobile app known as DuoMobile. This notion of security is built on the premise that users are intimately tied to their mobile phones, and those who try to log in to CAS can only be authenticated with possession of the phone.
However, this is preventative, or at the very least troublesome, for many users because DuoMobile is not an app that comes pre-installed on mobile phones. Therefore, users must download the app and will only use it when logging on — the app serves no other purpose. In fact, DuoMobile and other Two-Factor Authentication applications have various bugs that negatively affect the user experience, making a process as simple as logging in very complex.

To make the Two-Factor Authentication process more seamless while preserving security, it would be useful to integrate the authentication with an existing secure application with many users. Thus, my project aims to create a Two-Factor Authentication system using Facebook Messenger. We can leverage the incredibly wide user base of Facebook Messenger to make the Two-Factor Authentication much less painful. Indeed, this past summer, Facebook announced that Messenger has hit over 1 billion users. Thus, Two-Factor Authentication using Messenger would provide much less overhead than having to download an entirely separate application that serves no other functions. On the contrary, integration with Messenger would also help Messenger become a more comprehensive, one-stop-shop application.

To facilitate Two-Factor Authentication in Messenger, we can make use of Facebook bots. In the beginning of the calendar year, Facebook released development
support for their Messenger platform by announcing an API for Facebook Messenger Bots. These bots have increasingly become an integral part of the Messenger ecosystem. Companies have been using Messenger bots to handle transactions with users, play games, and follow news, among a myriad of other functions. The Application Programing Interface (API) of these bots contains capability to send and receive text. Facebook also provides support for natural language assistance. Thus, with respect to Two-Factor Authentication, these bots can also play an important role. By allowing bots to message users for authentication, this provides for high scalability. In addition, the use of Messenger bots still provides the second layer of security for Two-Factor Authentication because users are intimately tied to their Facebook and Facebook Messenger accounts, which are securely handled by Facebook.

My goal is to build an end-to-end login system that uses Two-Factor Authentication with Facebook Messenger Bots. Users would login, and my bot would prompt the user for verification on Facebook Messenger. I would also like to be able to export my Bot for use not only in my login system, but for any public login system that wishes to provide Two-Factor Authentication through Messenger. Through the process, I also want to learn about web development in Flask, a back-end Python web framework. I also want to learn how to use the Messenger Bot API and parse user input intelligently. Thus, my list of deliverables include the following:
- End-to-end login system built in Flask
- Server setup
- Facebook Messenger Bot approved by Facebook
- Exportable API for integration with any login system