TEAM FAT STACKS

CorgKey Demo Presentation

Adam Yu, Arjun Singh, Hunter Massey, Olivia Gillam, Simon Yu

SPONSOR: Allthenticate

MEET THE TEAM

Adam Yu (Lead)
Cross-Platform Dev

Simon Yu
Mobile Dev

Olivia Gillam
Mobile Dev

Arjun Singh
Authentication

Hunter Massey (Scribe)
Authentication

THE PROBLEM: PASSWORDS

NOT Secure

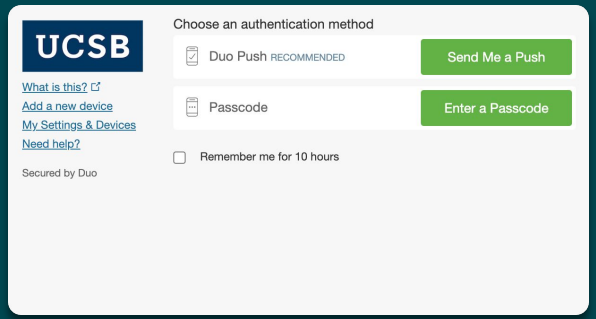
- Can be stolen by malicious actors through phishing attacks and data breaches
- Can be bypassed via SIM swapping & other types of attacks

NOT Convenient

- Users burdened with password memorization & management
- Password restrictions (e.g. requiring digits, caps, symbols) make the process of coming up with a password cumbersome

NOT Fast

- Two-Factor Authentication (2FA) adds security at the cost of convenience
- Time-consuming process of resetting a password



ABOUT 2FA…

According to Uber, having obtained the contractor's password, the hacker sent repeated log-in requests to the contractor's account and was then able to bypass Uber's two-factor log-in authentication—a system where a user is granted access after electronically confirming their identity twice—when the contractor finally accepted the authentication. The hacker was also admitted to the Uber Slack account and posted a message that read: "I announce I am a hacker and Uber has suffered a data breach."

https://www.bu.edu/articles/2022/what-you-need-to-know-about-uber-data-breach/

OUR SOLUTION: REPLACE PASSWORDS

- **CorgKey is a mobile application that acts as a mobile roaming FIDO authenticator**
  - CorgKey combines the **efficiency** of SSO and **security** of 2-factor authentication, while **minimizing the annoyance** that usually comes with combining those two
  - Unlike Duo, which is a secondary authenticator, CorgKey is a **primary authenticator**; simply having a connection between your mobile device and your browser will allow you access to your accounts
  - CorgKey implements the secure **FIDO2 protocol** using the **WebAuthn standard**

fido™ ALLIANCE | simpler stronger authentication

webauthn

USER FLOW

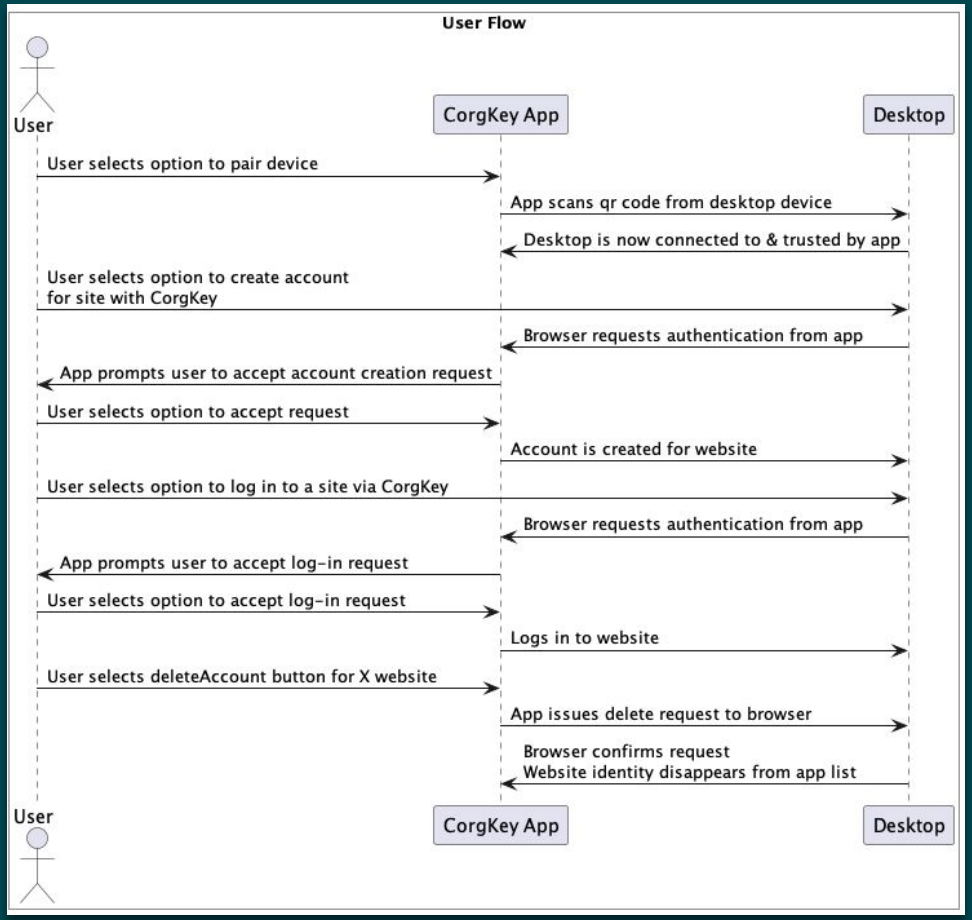
1. Using the CorgKey app, user pairs their phone with their desktop

2. Upon their first visit to a website on their computer, users will be prompted to create an account on their mobile device without coming up with a password

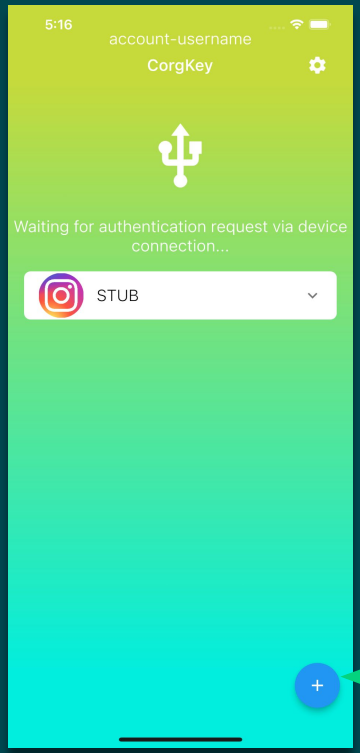
3. Upon future visits, users can log into the website by authorizing the attempt on their phone, rather than manually entering their username and password

4. **PROFIT**

USER SEQUENCE DIAGRAM

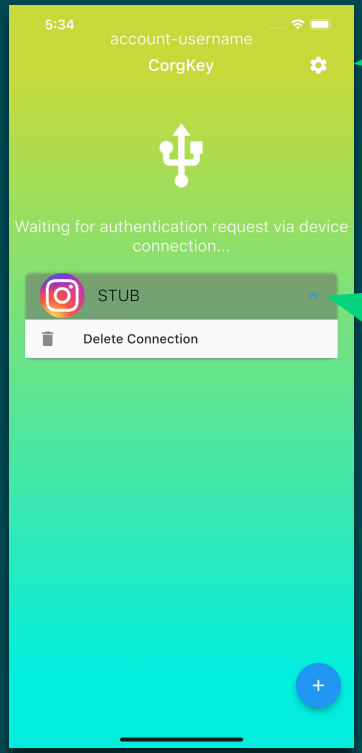


UI Overview



Accounts show up in a ListView on the main screen

(This button is just for testing)

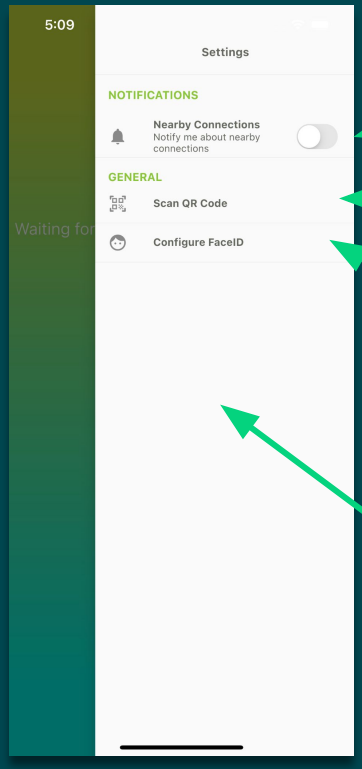


Open Settings drawer

Each account tile will have the options to

- delete account
- edit biometric requirements

UI Overview



Toggle notifications

Click to open camera to scan QR code for account creation

Click to bring up system preferences for allowing app to use face-ID (and/or fingerprint) biometric

TODO: Section to add/edit trusted devices (to allow for automatic logins via BLE)
(So on logins, you wouldn't even need to touch your phone)

CHALLENGES

Turning Phone into an Authenticator

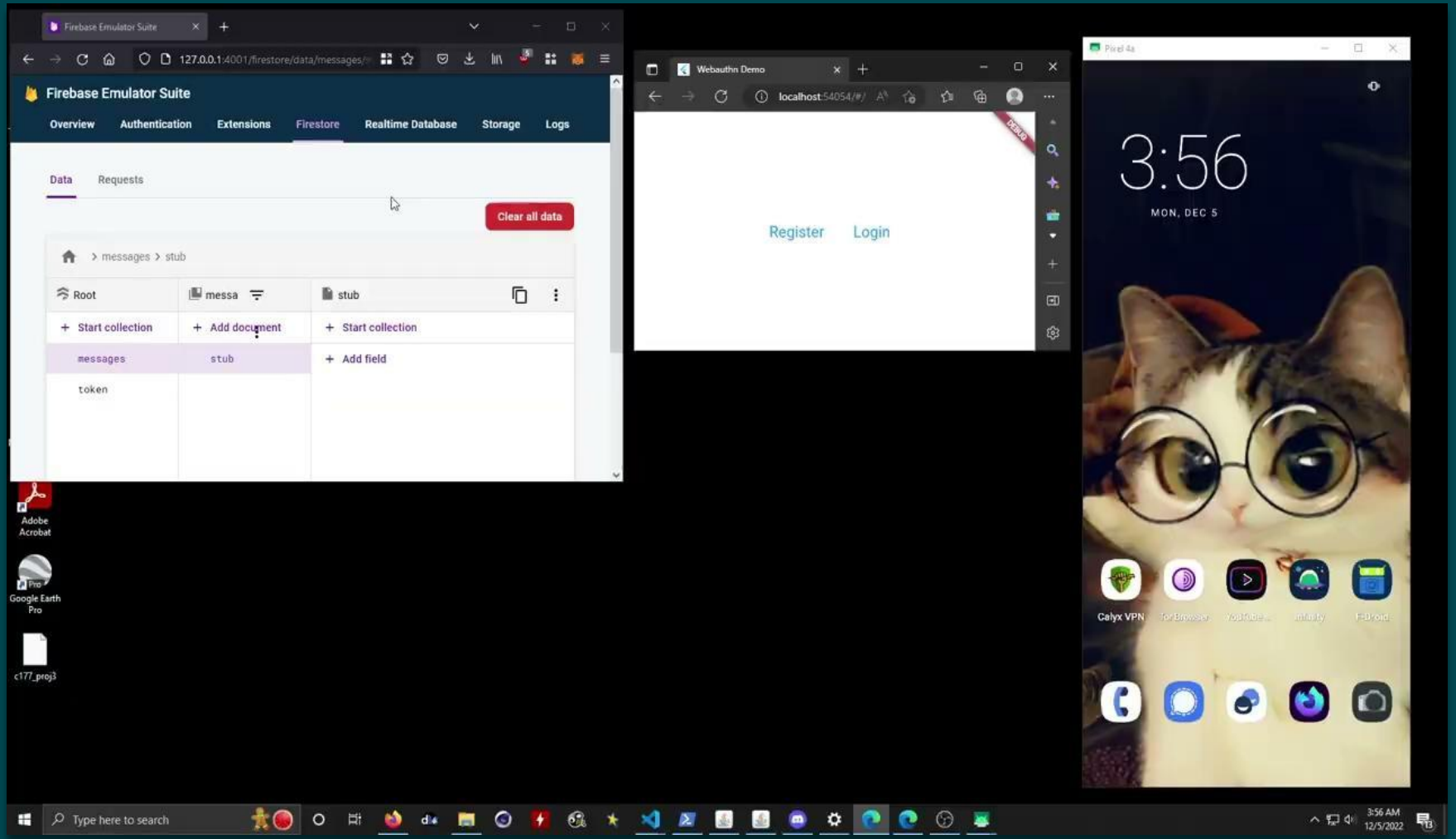
- **Mobile devices are not intrinsically hardware authenticator devices**
  - **Hack** — emulate a hardware authenticator device on the desktop
  - Have the mobile device tell it what to do
- **Increase in scope of project:**
  - Implement a desktop client that receives commands from the mobile device
  - Emulate a USB Hardware Authenticator device within the desktop client
  - Establish a secure pairing process to pair up mobile devices with desktop client

Phone—Desktop—Browser Communication

- **Initially wanted to use USB**
  - Compatibility issues w/ iOS/Lightning connector
  - Phone had to be connected to desktop w/ cable, inconvenient
- **Pivot to Firebase Cloud Messaging (FCM)**
  - After being paired, messages can be sent from one device to another using an FCM token
  - Intermediary step: Device A sends messages to a Firestore database, DB uses cloud function to send message to Device B
  - Advantage: it's **wireless!**

UTILIZED TECHNOLOGIES

VIDEO DEMO


Firebase Emulator Suite
127.0.0.1:4001/firestore/data/messages/
Firebase Emulator Suite
Overview
Authentication
Extensions
Firestore
Realtime Database
Storage
Logs
Data
Requests
Clear all data
messages
stub
Root
messa
stub
Start collection
Add document
Start collection
messages
stub
Add field
token
Webauthn Demo
localhost:54054/#/
DEMO
Register
Login
Pixel 4a
3:56
MON, DEC 5
Calyx VPN
Adobe Acrobat
Google Earth Pro
c177_proj3
Type here to search
3:56 AM
12/5/2022

WHAT'S NEXT?

Flesh out Mobile Application

- Implement planned features
  - Biometric/PIN authentication
  - Local storage of private keys
  - QR code scanning
  - Customizable permissions per connection
  - Persistent storage

Fully Implement Virtual Authenticator

- Use Human Interface Device (HID) specification to emulate hardware authenticator
  - Integrate virtual HID within desktop client

Develop Pairing Process Between Mobile/Desktop

- Generate QR codes that communicate FCM tokens between devices

Forward WebAuthn data through our pipeline

- Challenge + Relying Party ID sent to desktop through WebAuthn API calls
- Forwarded to App, which creates credential from private key

THANK YOU!