Authentication using Biometrics
Biometrics

- Passwords tough to remember, manage
- Many users have simple passwords (e.g. 1234) or do not change passwords
- Biometrics are unique physiological attributes of each person
  - Fingerprint, voice, face
- Can be used to replace passwords
  - No need to remember anything. Just be you. Cool!!
Android Biometric Authentication: Fingerprints

- **Fingerprint**: On devices with fingerprint sensor, users can enroll multiple fingerprints for unlocking device
Samsung Pass: More Biometrics

- **Samsung pass**: Fingerprint + Iris scan + facial recognition

- Probably ok to use for Facebook, social media

- Spanish bank BBVA’s mobile app uses biometrics to allow login without username + password

- Bank of America: pilot testing iris authentication since Aug 2017
Continuous Passive Authentication using Behavioral Biometrics
User Behavior as a Biometric

- User behaviors patterns are unique personal features. E.g
  - Each person’s daily location pattern (home, work, places) + times
  - Walk pattern
  - Phone tilt pattern

- **General idea:** Continuously authenticate user as long as they behave like themselves

- If we can measure user behavior reliably, this could enable **passive authentication**
BehavioMetrics
Ref: Zhu et al, Mobile Behaviometrics: Models and Applications

- Derived from Behavioral Biometrics
  - Behavioral: the way a human subject behaves
  - Biometrics: technologies and methods that measure and analyzes biological characteristics of the human body
    - Fingerprints, eye retina, voice patterns

- BehavioMetrics:
  - Measurable behavior to recognize or verify a human’s identity
Mobile Sensing → BehavioMetrics

- **Accelerometer**
  - Activity & movement pattern, hand trembling, driving style
  - sleeping pattern
  - Activity level, steps per day, calories burned

- **Motion sensors, WiFi, Bluetooth**
  - Indoor position and trajectory.

- **GPS**
  - outdoor location, geo-trace, commuting pattern

- **Microphone, camera**
  - From background noise: activity, type of location.
  - From voice: stress level, emotion
  - Video/audio: additional contexts

- **Keyboard, taps, swipes**
  - User interactions, tasks ....

• Network Factors
• Personal Factors
• Behavioral Factors
• Application Factors
BehavioMetrics → Security

- Track smartphone user behavior using sensors

- Continuously extract and classify features from sensors = Detect contexts, personal behavior features (pattern classification)

- Generate unique pattern for each user

- **Trust score**: How similar is today’s behavior to user’s typical behavior

- Trigger authentication schemes with different levels of authentication based on trust score
Continuous n-gram Model

- User activity at time $i$ depends only on the last $n$-1 activities
- Sequence of activities can be predicted by $n$ consecutive activities in the past
  \[ P(l_i | l_{i-n+1}, l_{i-n+2}, \ldots, l_{i-1}) \quad \text{or} \quad P(l_i | l_{i-n+1}^{i-1}) \]
- Maximum Likelihood Estimation from training data by counting:
  \[ P_{\text{MLE}}(l_i | l_{i-n+1}^{i-1}) = \frac{C(l_{i-n+1}, \ldots, l_{i-1}, l_i)}{C(l_{i-n+1}, \ldots, l_{i-1})} \]
- MLE assign zero probability to unseen n-grams
Classification

- Build $M$ BehavioMetrics models $P_0, P_1, P_2, \ldots, P_{M-1}$
  - Genders, age groups, occupations
  - Behaviors, activities, actions
  - Health and mental status

- Classification problem formulated as

$$\hat{u} = \arg\max_m P(L, m) = \arg\max_m \sum_{i=1}^{N} \log P_m(l_i | l_{i-n+1}^{i-1})$$
Anomaly Detection Threshold

![Diagram showing average log probability over sliding window position with threshold levels indicated at C, D, A, B.](Image)
Behavioral Biometrics Issues: Shared Devices
BehavioMetric Issues: Multi-Person Use

- Many mobile devices are shared by multiple people
  - Classifier trained using person A’s data cannot detect Person B

- **Question:** How to distinguish when person A vs person B using the shared device

- How to segment the activities on a single device to those of multiple users?
BehavioMetric Issues: Multi-Device Use

- Many people have multiple mobile devices
  - Classifier trained on device 1 (e.g. smartphone) may not detect behavior on device 2 (e.g. smartwatch)
  - **Question:** How to match same user’s session on multiple devices
    - E.g. Use Classifier trained on smartphone to recognize user on smartwatch
  - How to match user’s activity segments on different devices?

---

![Diagram showing user activity across multiple devices over time]
ActivPass
Passwords are mostly secure, simple to use but have issues:

- Simple passwords (e.g. 1234): easy to crack
- Secure passwords hard to remember (e.g. $emime)$@(*$@)9
- Remembering passwords for different websites even more challenging
- Many people use same password on different websites (dangerous!!)
Unique human biometrics being explored

Explicit biometrics: user actively makes input
- E.g. finger print, face print, retina scan, etc

Implicit biometrics: works passively, user does nothing explicit to be authenticated.
- E.g. unique way of walk, typing, swiping on screen, locations visited daily

This paper: smartphone soft sensors as biometrics: calls, SMS, contacts, etc

Advantage of biometrics: simple, no need to remember anything
**ActivPass Vision**

- **Observation:** rare events are easy to remember, hard to guess
  - E.g. A website user visited this morning that they rarely visits
    - User went to CNN.com today for the first time in 2 years!
  - Got call from friend I haven’t spoken to in 5 years for first time today

- **Idea:** Authenticate user by quizzing them to confirm rare (outlier) activities
  - What is caller’s name from first call you received today?
  - Which news site did you not visit today? (CNN, CBS, BBC, Slashdot)?
ActivPass Vision

- Authentication questions based on outlier (rare) activities generated from:
  - Call logs
  - SMS logs
  - Facebook activities
  - Browser history
ActivPass Envisioned Usage Scenarios

- Replace password hints with Activity questions when password lost
- Combine with regular password (soft authentication mechanism)
- Prevent password sharing.
  - E.g. Bob pays for Netflix, shares his login details with Alice
How ActivPass Works

- Activity Listener runs in background, logs
  - Calls, SMS, web pages visited, etc

- When user launches an app:
  - Password Generation Module (PGM) creates $n$ password questions based on logged data
  - If user can answer $k$ of password questions correctly, app is launched!
ActivPass Vision

- User can customize
  - Number of questions asked,
  - What fraction of questions $k$ must be answered correctly
  - Question format
  - Activity permissions

<table>
<thead>
<tr>
<th>Question formats</th>
<th>Example questions asked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>Have you received a call from Alice at around 10 pm on 19/09/2014?</td>
</tr>
<tr>
<td>MCQ</td>
<td>Please write the options of the links you visited, this week in comma separated way (Ex: A, B): A. CNN; B. BBC; C. SKY News; D. Reuters</td>
</tr>
<tr>
<td>Text</td>
<td>Whom did you call at around 7 pm on 17/09/2014? ?</td>
</tr>
<tr>
<td></td>
<td>Hint: (AI*)</td>
</tr>
</tbody>
</table>

- Paper investigated ActivPass utility by conducting user studies
How ActivPass Works

- Periodically retrieves logs in order to classify them using **Activity Categorization Module**
  - Tries to find outliers in the data. E.g. Frequently visited pages vs rarely visited web pages
### ActivPass: Types of Questions Asked Vs Data Logged

<table>
<thead>
<tr>
<th>Source</th>
<th>Details of data collected</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>Time, Receiver/Sender Name</td>
</tr>
<tr>
<td>Call</td>
<td>Time, Type (incoming, outgoing), Name of other person, Duration</td>
</tr>
<tr>
<td>Audio</td>
<td>Title of Music added in this week, Alarm tone, Ring tone</td>
</tr>
<tr>
<td>Web</td>
<td>URL, Time of visit</td>
</tr>
<tr>
<td>Link visited from</td>
<td>URL, Time of visit</td>
</tr>
<tr>
<td>Facebook</td>
<td>Name of Private (secret and closed) groups</td>
</tr>
<tr>
<td>Facebook Group</td>
<td>Name of pages created by user</td>
</tr>
<tr>
<td>Facebook Pages</td>
<td>Name of Facebook friends of user</td>
</tr>
<tr>
<td>Facebook Message</td>
<td>Time (in milliseconds from epoch), Name of other person, Msg Id, Thread Id</td>
</tr>
</tbody>
</table>
ActivPass: Evaluation

- Over 50 volunteers given 20 questions:
  - Avg. recall rate: 86.3% ± 9.5 (user)
  - Avg guessability: 14.6% ± 5.7 (attacker)

- Devised Bayesian estimate of challenge given $n$ questions where $k$ are required

- Tested on 15 volunteers
  - Authenticates correct user 95%
  - Authenticates imposter 5.5% of the time (guessability)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k$</th>
<th>Authentic user</th>
<th>Impostor</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>0.554</td>
<td>0.0004</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.906</td>
<td>0.011</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.989</td>
<td>0.1043</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.998</td>
<td>0.468</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.642</td>
<td>0.0031</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td><strong>0.948</strong></td>
<td><strong>0.0577</strong></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.996</td>
<td>0.3771</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.745</td>
<td>0.0213</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.981</td>
<td>0.2707</td>
</tr>
</tbody>
</table>

Optimal $n$, $k$
Smartphones + IoT Security Risks
Cars + Smartphones → ?

- Many new vehicles come equipped with smartphone integration / capabilities in the infotainment system (Android Auto!)
Smartphones that Drive

If a mobile app gets access to a vehicle’s infotainment system, is it possible to get access to (or even to control) driving functionality?
Smart Vehicle Risks

- Many of the risks and considerations that we discussed in this course can be applied to smart vehicles and smartphone interactions.

- However, many more risks come into play because of the other functionality that a car has compared to a smartphone.
Secure Mobile Software Development Modules
Introduction

● Many Android smartphones compromised because users download malicious software disguised as legitimate apps

● Malware vulnerabilities can lead to:
  ● Stolen credit card numbers, financial loss
  ● Stealing user’s contacts, confidential information

● Frequently, unsafe programming practices by software developers expose vulnerabilities and back doors that hackers/malware can exploit

● Examples:
  ● Attacker can send invalid input to your app, causing confidential information leakage
Secure Mobile Software Development (SMD)

- **Goal**: Teach mobile (Android) developers about backdoors, reduce vulnerabilities in shipped code

- **SMD**:
  - Hands-on, engaging labs to teach concepts, principles
  - Android plug-in: Highlights, alerts Android coder about vulnerabilities in their code
  - Quite useful
SMSD: 8 Modules

- Focussed more on teaching you about the modules
- M0: Getting started
- M1: Data sanitization for input validation
- M2: Data sanitization for output encoding
- M3: SQL injections
- M4: Data protection
- M5: Secure inter-process communication (IPC)
- M6: Secure mobile databases
- M7: Unintended data leakage
- M8: Access control

https://sites.google.com/view/projectsmsd/home
Open Source SMSSD API Plugin for Android Studio IDE

- Plugin you can use to scan your Android projects for vulnerabilities

- M0. Getting Started with SpotBugs for Android Static Code Analysis
- M1. Potential SQL Injection Vulnerability Detecting with SpotBugs
- M2. Data Sanitization for output encoding Vulnerability Detecting with SpotBugs
- M3. Intent Interception and Spoofing Vulnerability Detecting with SpotBugs
- M4 InterAppSender Access Control Vulnerability Detecting with SpotBugs
M7 & M8 Overview

- **M7: Blah**
- **Unintended Data Leakage**
  - Understand fundamental concepts of unintended data leakages from the clipboard
  - Understand defenses against these unintended data leakages

- **M8: Inter-App Secure IPC vulnerabilities**
  - Malicious app can exploit security loophole in Broadcast Receivers to intercept valuable information