A keyboard that manages your passwords in Android

PRISMS 2014
Aalborg, 11-14 May 2014

Faysal Boukayoua
Vincent Naessens
Bart De Decker

KU Leuven
Overview

- Motivation
- Approach
- Implementation
- Other considerations
- Evaluation
- Conclusion
Motivation: passwords

Poor usability

Insecure
Motivation: passwords on mobile devices

Typing inconvenience

“There’s an app for everything”
**Motivation: widely used mobile solutions**

<table>
<thead>
<tr>
<th></th>
<th>Platform-based account mgmt</th>
<th>Browser pwd mgmt</th>
<th>Password vaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure provisioning?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Disruptive to workflow?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Support for all passwords?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Changes to app?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Portable to other platforms?</td>
<td>Yes, but different APIs</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Approach: key concepts

- Passwords through the keyboard
- Secure password storage
- App authentication
- User authentication
Approach: architecture

- User
- MDM server
- Relying app 1
- Relying app 2
- Relying app 3
- Keyboard
- Password store
- Password manager
- Mobile device
- Mobile OS
- System keystore
**Approach: protocol**

1. Log in
2. Prompt for user name and password
3. Request user name and password
4. Get account(s) for relying app
5. Return account(s)
6. Display accounts
7. Select account
8. Get account
9. Return user name and password
10. Prompt to indicate corresponding field
11. Indicate corresponding field
12. Populate field

**If [multiple accounts]**

Repeat for user name, password
Implementation: prototype

- Google Nexus 4
- Android 4.3
- All components in 1 app package
- Configure in *Language and Input*

Retrieve user name and password
Implementation: prototype

- Google Nexus 4
- Android 4.3
- All components in 1 app package
- Configure in *Language and Input*

Store user name and password
Implementation: prototype

- Google Nexus 4
- Android 4.3
- All components in 1 app package
- Configure in *Language and Input*

Generate strong password (auxiliary)
Implementation: retrieving an account
Implementation: retrieving an account
Implementation: retrieving an account
Implementation: retrieving an account

Please indicate the location of the password field.

- test2@skype
- Password

Sign in
Implementation: retrieving an account
Implementation: storing an account
Implementation: storing an account
Implementation: storing an account
Implementation: storing an account
Implementation: storing an account

Please indicate the location of the user name field.

prismstest

Password field
Implementation: storing an account
Implementation: storing an account
Implementation: storing an account
Implementation: keyboard

Android Input Method Framework (IMF)

- Strict separation between
  - client apps
  - client apps and editor (IME)
- Only one client *active* at once
- IME change only by user, not app
- Not just keyboards: voice, handwriting,…
Implementation: password store

- Symmetric encryption
- Protect crypto key using KeyChain
  - Android 4.3+: hardware-backed RSA key storage
  - App-level credential access
- Alternative: symmetric key in secure element
Implementation: user authentication

- System passcode
- Android Device Administration API
  - Force enabled passcode
  - Strength requirements
  - Max inactivity for lock screen
- Others:
  - Max failed attempts
  - Expiry
  - Password history restrictions
  - ...
Implementation: app authentication

- Access control to app passwords
- Compound app ID
  - Package name (e.g. `com.skype.raider`)
  - Developer signature
- Extension: password pooling
  - Why?
    - Same authentication infrastructure
    - Browsers
  - How?
    - Same signature
    - User-composed lists
    - ...
Other considerations

• Subdivision for website passwords
  - Android app
    - Website domain

• Mobile Device Management
  - Password policies
  - Password pooling between apps
  - Require hardware-backed KeyChain
  - Application white-/blacklists

• Backups: recovery from loss
• Synchronisation: multi-device access
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Platform-based account mgmt</th>
<th>Browser pwd mgmt</th>
<th>Password vaults</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure provisioning?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Disruptive to workflow?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Support for all passwords?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Changes to app?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Portable to other platforms?</td>
<td>Yes, but different APIs</td>
<td>Yes</td>
<td>Yes</td>
<td>Currently only Android</td>
</tr>
</tbody>
</table>
Conclusion

• Passwords are:
  o here to stay… for now
  o more cumbersome on mobile devices

• Contributions
  o Interoperability
    • No platform or app changes
    • Support for all passwords
  o Usability: integration in user’s workflow
  o Secure provisioning and storage