Windows Phone 7 Picturelock Application

Team 2

<Student's names redacted>
Contents

- Demo
- Access Control
- Threats and Vulnerabilities
- Application Description
- Pitfalls and Future features
Demo
Access Control: Types

- Physical
  - Locks
  - Fences
  - Security Guards
- Administrative
  - Policies and procedures
  - Training
- Technical
  - Software and hardware devices
Access Control: AAA

- Identity and Authentication/Authorization/Accountability
Access Control: Process Flow

1. Request Access
2. Identify
3. Authenticate
4. Authorize
Authentication: Categories

Composed of three categories
- Something you know
  - Passwords
  - Passphrases
- Something you have
  - Magnetic key card
  - Smart card
  - Token device
- Something you are
  - Fingerprint
  - Retinal Scan
Authentication: Passwords

- The cheapest/easiest form of authentication
- Works well with most applications
- Also the weakest form of access control
  - Prone to a number of attacks
- Requires administrative controls to be effective
  - Minimum length/complexity
  - Password aging
  - Limit failed attempts
Authentication: Smart Cards/Security Tokens

- More expensive/harder to implement
- Prone to the vulnerability of loss or theft
- Very strong when combined with another form of authentication
  - Multi-factor Authentication
- Does not work well in all applications such as smart phones
Authentication: Biometrics

- More expensive/harder to implement
- Prone to error
  - False Accept Rate/False Reject Rate
- Strong authentication when it works
- Does not work well in all applications
  - Fingerprint readers becoming more common (Atrix 4g)
## Authentication: Comparison

<table>
<thead>
<tr>
<th></th>
<th>Passwords</th>
<th>Smart Cards</th>
<th>Biometrics</th>
<th>Picture Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Weak</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Easy</td>
<td>Medium</td>
<td>Hard</td>
<td>Medium</td>
</tr>
<tr>
<td>Implementation</td>
<td>Easy</td>
<td>Hard</td>
<td>Hard</td>
<td>Medium</td>
</tr>
<tr>
<td>Works for phones</td>
<td>Yes</td>
<td>No</td>
<td>Possible</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Threats to Traditional Passwords

- Shoulder Surfing
- Brute Force Attacks
- Dictionary Attacks
- Forgotten Passwords
- Cracking
- Social Engineering
Shoulder Surfing

- An attacker watches as a legitimate user performs some kind procedure required for authentication.
- Effectiveness depends on ability to observe and reproduce actions.
Brute Force Attacks

- Attacker submits random passwords in hope of finding a correct one or to exhaust all possible combinations.
- Effectiveness depends on number of possible passwords and number of attempts allowed or time given to attacker.
Dictionary Attack

- Uses a list of common passwords to attempt to guess a correct password.
- Effectiveness based on strength of password, and therefore are based on constraints placed on what passwords can be created.
Forgotten Passwords

- If a user forgets a password, then he or she will be denied access to their resources.
- Passwords can usually be recovered, but if the recovery system has been compromised, then this creates a new vulnerability (compromised email address)
Cracking

- If an attacker gains access to a password hash, then he or she may attempt to use it to determine a user’s password.
- Effectiveness depends on backend implementation.
Social Engineering

- An attacker attempts to persuade a user to give information, materials, etc. required to access a system.
- Effectiveness based on many factors,
# Vulnerabilities

<table>
<thead>
<tr>
<th></th>
<th>Passwords</th>
<th>Smart Card</th>
<th>Biometrics</th>
<th>Picture Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic Monitoring</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Should surfing</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low to High</td>
</tr>
<tr>
<td>Brute Force</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low to Medium</td>
</tr>
<tr>
<td>Dictionary</td>
<td>Low to High</td>
<td>N/A</td>
<td>N/A</td>
<td>Low</td>
</tr>
<tr>
<td>Social Engineering</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Picture Lock

• **Traditional Security**
  • User selects a password string of characters from a alphabet. That string must be entered to gain access.

• **Attack 1**
  • Snooping and Brute force

• **Counter Measure**
  • Obfuscate the alphabet to make it difficult to understand what characters are being presented and pressed.
  • Obfuscate the password entry to make it difficult for brute force to sequence through its attack
Picture Lock

Picture “Alphabet”

- Passwords will be defined as a sequence of picture “characters”. The characters will be taken from an “alphabet” of theme images – with multiple representations for each theme. e.g. row 1 = cat, row 2 = cow, row 3 = dog, row 4 = duck
User Login

- A security manager will present a grid that is randomly populated with the alphabet’s characters – each a random representation of the given character. The following shows two different example login screens for our 4 character sample alphabet:
**Picture Lock**

- **Attack 2**
  - Automated brute force and snooping can use image analysis (e.g. Google Goggles) to identify what the pictures are and learn the theme characters.
  - e.g. \{cat, cow, dog, duck\}.

- **Counter Measure**
  - Character themes should be based on knowledge that only the intended user would likely know. Themes should be coordinated so that, when looking at the entire set of images, it is difficult for someone else to guess which images are likely grouped together into “characters”.
  - e.g. trip themes, family themes, secret themes, etc.
Attack 3

Since the given login displays images of all alphabet characters at the same time, attacks can learn that the given images are all of different themes – therefore if picture ‘A’ is never shown with picture ‘B’, then they are likely in the same theme.

Counter Measure

Don’t display the entire alphabet and display duplicate characters. Only show the password characters plus a random selection from the remaining alphabet characters.

Note: displaying duplicates increases odds of guessing. Security manager must have some knowledge of password to know what characters to offer.
Picture Lock

• Attack 4
  • Snooping can simply record the images used for login over time. e.g. image set 1 is clicked first, set 2 is clicked next, etc. – the attack would then just look for images in the sets.

• Counter Measure
  • The password length and theme size should be as large as practical. Since repeated snooping could learn set images fairly quickly, the character’s set of theme images should be dynamic. Displayed images should periodically be removed from the theme and replaced with new ones.
  • If a theme image is used only once, then there is no chance for snooping to reuse images or create login character theme sets - but this would require frequent additions to theme image collections.
Picture Lock

Typical Use Case:

1. User provides identity to security manager
2. Mgr. provides user with set of images
3. User passes a selected image to Mgr.
4. Mgr. tries to decrypt data with set of passed chars from user
5. If decrypt, accept login – else, pass replacement image
6. Repeat 3-5 until access or reset to start over

Security Manager
- Set of Characters containing password (password itself?)
- Set of Characters not in password
- Set of characters returned by user

Encrypted User Data
Pitfalls and future features of PictureLock and security organizer

- What happens if you forget what your login is?
  - Currently there is no way to recover this
  - Having a recovery method defeats the purpose of picture lock
How do you protect pictures and picture lock password?

- Picture lock's pictures needs to be encrypted
- Security organizer’s data is already Encrypted
- There is a pitfall in that picturelock needs to know the first character of the password in order to ensure its displays the picture
• Instead the process could behave like a client and server authentication system:

  • enter a password, client encrypts it and send it off to a server, server verifies password and grants permission to the client to start the security organizer

  • This client server communication would need to be secure and server’s data needs to be secure
Future feature: Allow user to create multiple databases

- Create new
- Open existing
- Delete
  - Confirm Delete
• Future feature: Have a way to have user maintain and update databases
  • Finish implementing Uploading and Deleting pictures/Themes
  • Format pictures
    • Brightness
    • Contrast
    • Rotate
    • Save changes
    • Cancel Changes
• Future feature: Shuffle pictures half way through password entry in case someone is watching it over your shoulder

• Have pictures as NULL so they don’t add to the password typed in

• Picture lock could work better as a lock screen feature for a phone
Future feature: Intrusion Detection System
- Detect when someone is trying to guess passwords
- Limit amount of attempts before timed lockout
• Future feature: Have a password change option
  • Enter current password and then re-enter new password twice to change it.
• Future feature: Links go to websites when clicked
• Eliminate bugs
References