An Anti-Phishing mechanism for Single Sign-On based on QR-Code

Syamantak Mukhopadhyay
&
David Argles

School of Electronics & Computer Science
University of Southampton
Introduction

- Internet & Web 2.0
  - User-centric services
  - Services available Online.
  - Most services require username/password for authentication & authorization
    - Too many of them to remember (25 on an average)
    - Use same password!! -> Password fatigue
- Single Sign-On to the rescue
Single Sign-On

- One ring to rule them all!

- Shibboleth
  - Uses SAML
  - Best suited for portal or Intranet applications

- OpenID
  - User can chose his/her Identity provider
  - No pre-established contract required between Service Provider and Identity Provider

- Information Card & MS Cardspace
  - Different Identity sectors for different purposes.
  - Identity sectors are stored in client machine!!
Single Sign-On process

1. User requests for service

2. User is redirected to IdP

3. User provides identity to IdP

4. IdP Sends Security assertion to Service provider

5. User Accesses the service
Phishing & Single Sign-On

Account compromised

Phishing Page
Previous works on anti phishing

- Client side (Browser solutions)
  - Personal icon from myOpenID
  - VeriSign - Validation Certificate for IE7 and seatbelt for Firefox
- Use two passwords – Based on Kerberos
  - Show two phishing page instead of one!!
- Use mobile SIM in authentication
- For each login generate a token and send it to the user as email
  - breaks SSO, user needs to login to open email first -> Single Identity Sign On (SISO)
- Use I-PIN
  - Can’t be implemented globally
Proposed Model

- Avoid passwords when accessing a service
  - Use QR-Code to generate one time password

- Based on the assumption that most internet users are equipped with a mobile device that has a camera.

- Uses two phase approach
  - User registration phase
  - User verification phase
User Registration Phase

1. User Provides ID_A and RP_A

2. RP_A and Random number

2. Secret Key generation process using RP_A

3. IdP Returns X_A

User

Identity Provider

<table>
<thead>
<tr>
<th>ID_A</th>
<th>Username or identity of the User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP_A</td>
<td>Root password of the user</td>
</tr>
<tr>
<td>X_A</td>
<td>Secret key of the user</td>
</tr>
<tr>
<td>E_QR</td>
<td>Encoded QR code</td>
</tr>
<tr>
<td>D_QR</td>
<td>Decoded QR code</td>
</tr>
</tbody>
</table>
User Verification Phase

1. User requests Service

2. User is redirected to IdP

3. User provides IDA

4. Retrieve XA

5. Returns EQR and T1

6. Return DQR and T2

6. Mobile App to decode QR code

7. Return Security assertion

8. User accesses Service

User Identity

<table>
<thead>
<tr>
<th>IDA</th>
<th>Username or identity of the User</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP A</td>
<td>Root password of the user</td>
</tr>
<tr>
<td>X A</td>
<td>Secret key of the user</td>
</tr>
<tr>
<td>EQR</td>
<td>Encoded QR code</td>
</tr>
<tr>
<td>DQR</td>
<td>Decoded QR code</td>
</tr>
</tbody>
</table>
Proposed Model – User Interaction

Login with your Identity

Please Enter Your Username

Submit
Proposed Model – User Interaction

QR Code for Login

Take a picture with your Mobile

Enter Token

Please Enter the code from your Mobile

Submit

Or

Wait for automatic login
Proposed Model – User Perspective

User’s Action

App()

- Decode the QR Code
- If web enabled mobile
  - Send the decoded value using https
- Else display the decoded value to be entered manually.
- Users logs in!

Image Source:
Proposed Model – Key Points

- Generation of Secret key ($X_A$) is dynamic
  - $X_A$ is compromised – generate again
    - Reset root password

- Does not introduce any new complications in user verification phase

- Simple and usable
Proposed Model - Security Analysis

- **Phishing Attack**
  - Root password is never disclosed during verification phase.
  - Secret key is generated from Root password using one way hash.
    - Hence Root password can’t be derived from Secret key
  - If secret key is compromised, simply generate another one.

- **Other attacks**
  - QR-Code is generated using a random number
  - Decoded value uses Timestamp - accepted only within a small time limit
  - Fairly safe from both man in the middle attacks and replay attacks
Conclusion

- New SSO model with mobile QR code based onetime password schema
- Secure from phishing
- Prevents other attacks as well (replay & man in the middle)
- Simple from users perspective
- Can be substituted in any system that uses username/password
Thank You!

Questions ?