Tigress Working Group

Transfer digital credentials securely

July 27th, 2022
<table>
<thead>
<tr>
<th>Topic</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Statement</td>
<td>Casey</td>
</tr>
<tr>
<td>Goals</td>
<td>Casey</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>Casey</td>
</tr>
<tr>
<td>Use Cases</td>
<td>Alex</td>
</tr>
<tr>
<td>Requirements</td>
<td>Alex</td>
</tr>
<tr>
<td>Out of Scope</td>
<td>Alex</td>
</tr>
<tr>
<td>Q &amp; A</td>
<td>All</td>
</tr>
<tr>
<td>Proposed Solution</td>
<td>Dmitry</td>
</tr>
<tr>
<td>Stateless Workflow</td>
<td>Dmitry</td>
</tr>
<tr>
<td>Stateful Workflow</td>
<td>Dmitry</td>
</tr>
<tr>
<td>Q &amp; A</td>
<td>All</td>
</tr>
</tbody>
</table>
Authors and Contributors

- Dmitry Vinokurov
- Alex Pelletier
- Nick Sha
- Casey Astiz
- Matt Byington
- Matthias Lerch
- Ben Chester
- Jean-Luc Giraud
- Yogesh Karandikar
- Alexey Bulgakov
- Tommy Pauly
- Crystal Qin
- Adam Bar-Niv
- Manuel Gerster
- Igor Gariev
Problem
Problem Statement
Problem Statement

Today, no standardized method exists in a cross-platform, cross-vertical capacity that would enable users to share secure credentials.
Goals
Goals

Minimize Friction for Sharing
Goals

Minimize Friction for Sharing

Maintain Access Control
Goals

Minimize Friction for Sharing

 Maintain Access Control

Security and Privacy
Problem Vocabulary
Problem Vocabulary

Credential Authority
Problem Vocabulary

Credential Authority

Credential Information
Problem Vocabulary

- Credential Authority
- Provisioning Information
- Credential Information
Problem Vocabulary

- Credential Authority
- Provisioning Information
- Credential Information
- Sender Device
Problem Vocabulary

- Credential Authority
- Provisioning Information
- Receiver Device
- Credential Information
- Sender Device
Problem Vocabulary

- Credential Authority
- Provisioning Information
- Receiver Device
- Credential Information
- Sender Device
- Relay Server
Use Cases
Use Cases
Vehicle

Friend

Valet
Use Cases
Residential

Visitor

Housekeeper
Use Cases

Hotel

Partner

Friends
Requirements
Solution Expectations
Solution Expectations

- Allow share initiation over any channel
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
- Allow multiple round trip communications
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
- Allow multiple round trip communications
- Allow sender and recipient online at different times
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
- Allow multiple round trip communications
- Allow sender and recipient online at different times
- Allow opaque message content
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
- Allow multiple round trip communications
- Allow sender and recipient online at different times
- Allow opaque message content
- Allow a variety of types of credentials to be transferred
Solution Expectations

- Allow share initiation over any channel
- Allow share invitation preview
- Allow multiple round trip communications
- Allow sender and recipient online at different times
- Allow opaque message content
- Allow a variety of types of credentials to be transferred
- Allow management of share by Sender or Receiver
Security Goals
Security Goals

• Ensure only the intended recipient is able to provision
Security Goals

- Ensure only the intended recipient is able to provision
- Ensure credential can only be provisioned once (anti-replay)
Security Goals

- Ensure only the intended recipient is able to provision
- Ensure credential can only be provisioned once (anti-replay)
- Ensure sender has the intent to transfer
Privacy Values
Privacy Values

- Server should not be able to associate sender and receiver
Privacy Values

- Server should not be able to associate sender and receiver
- Server should not see shared content
Privacy Values

- Server should not be able to associate sender and receiver
- Server should not see shared content
- Server should not be able to intercept or redeem
Out of Scope
Out of Scope

- Mechanism for receiver to accept share with credential authority
Out of Scope

• Mechanism for receiver to accept share with credential authority
• Mechanism for sender to get provisioning information from credential authority
Out of Scope

• Mechanism for receiver to accept share with credential authority
• Mechanism for sender to get provisioning information from credential authority
• User Interface (UI) for sender or receiver
Out of Scope

• Mechanism for receiver to accept share with credential authority
• Mechanism for sender to get provisioning information from credential authority
• User Interface (UI) for sender or receiver
• Format or content of encrypted data
Technical Solution
Relay Server

- Establishes “connectivity” between sender and recipient
- Simple mailbox, decoupled from provisioning logic
- Only sees encrypted data and metadata
Stateless and Stateful flows

• In Stateless flow there is a single credential data transfer:
  • Sender -> Relay -> Receiver

• In Stateful flow there are multiple data transfers between Sender, Relay and Receiver to prepare credential data for registering or provisioning by Receiver
  • Sender -> Relay -> Receiver
  • Additional round trip between Receiver and Sender for new credential authorization
APIs

• Create Mailbox: POST /{version}/m
• Read Display Information from Mailbox: GET /{version}/m/{mailboxIdentifier}
• Read Secure Content from Mailbox: POST /{version}/m/{mailboxIdentifier}
• Update Mailbox: PUT /{version}/m/{mailboxIdentifier}
• Delete Mailbox: DELETE /{version}/m/{mailboxIdentifier}
Sharing Process

Credential Authority

Sender Device

Relay Server

Receiver Device
Stateless Sharing Process

1. Sender Device
   a. Sender initiates Wallet sharing process
   b. Sender configures recipient’s entitlements & capabilities

2. Device generates secret

Authorized share & get provisioning info

Credential Authority

Relay Server

Receiver Device
Stateless Sharing Process

1. Encrypt provisioning info
2. Deposit encrypted data
3. Result is a unique shareUrl
4. Send share URL with encryption key via SMS/email/WhatsApp/etc.

example.com/v1/m/12345#Secret
Stateless Sharing Process

1. User taps share URL

2. Credential Authority

3. Relay Server

4. Retrieve Display Information

5. Receiver Device

Sender Device

User taps share URL
Stateless Sharing Process

1. User taps share URL
2. Credential Authority
3. Relay Server
4. example.com/v1/m/12345#Secret
5. Receiver Device
6. Relay Server

a. Retrieve encrypted data
b. Decrypt provisioning info

c. Device

d. Device

- Example: example.com/v1/m/12345#Secret
- Authority: Credential Authority
- Devices: Sender Device, Relay Server, Receiver Device
Stateless Sharing Process

1. User taps share URL
2. Credential Authority
3. Provision credential information to device
4. Relay Server
5. Receiver Device
6. Delete mailbox
7. Redeem provisioning info and get credential info
8. Provision credential information to device
Stateful Sharing Process

1. **Sender Device**
   - a. Retrieve and decrypt provisioning info

2. **Relay Server**
   - 3a. Update Provisioning Information

3. **Receiver Device**
   - 4. Update Provisioning Information
   - 5. Receive and decrypt provisioning info

4. **Sender Device**
   - 6. Retrieve and decrypt provisioning info

5. **Relay Server**
   - 7. Update Provisioning Information

6. **Receiver Device**
   - 8. Update Provisioning Information
User taps share URL

Stateful Sharing Process

1. User taps share URL
2. Sender Device
3. Relay Server
4. Relay Server
5. Receiver Device
6. Relay Server
7. Relay Server
8. Relay Server
9. Relay Server
10. Relay Server

- a. Update Provisioning Information
- b. Updated provisioning info

a. Retrieve and decrypt
Ending Notes

- Tigress Problem and Solution.
- Goals.
- IETF adoption.
- https://datatracker.ietf.org/doc/draft-secure-credential-transfer/
Thank you!
Q&A