TIGRESS Interim Meeting

February 23, 2023

Authors: Brad Lassey, Casey Astiz, Alex Pelletier, Dmitry Vinokurov, James Willcox

Agenda

- Review of work since IETF 115
- Updates to Requirements
- Updates to Threat Model
- Quick Review of new documents

Review of work since IETF 115

- Addressed feedback about Requirements and Threat Model documents
- Created 1-pagers on alternate suggested approaches for TIGRESS

Updates to Requirements

- https://datatracker.ietf.org/doc/draft-tigress-requirements/
- Removed user interface related requirements
- Altered requirements to be more generic and address feedback from working group
- Added an example diagram for context

Requirements - Hopefully uncontroversial

- (Req-XPlatform) Solution shall support transfer of digital credential across different platforms (e.g. from Android to iOS).
- (Req-CredentialType) The solution shall support transfer of various digital credential types, based on symmetric and asymmetric cryptography, public and proprietary standards.
- (Req-Security) Solution should provide security of the provisioning data transferred (confidentiality, integrity and availability of provisioning information in transit).
- (Req-Connectivity) Sender and Receiver shall be allowed to be online at different times. Sender and Receiver shall not need to be online at the same time. This requirement allows devices to connect to network to only exchange the portion of information required during the transfer, allowing them upload or download data in turns to network servers.
- (Req-RoundTrips) Solution shall allow for multiple data exchanges between sender and receiver devices in the process of credential transfer. This requirement shall align with (Req-Connectivity) above.
- (Req-Opaque) In the case when an intermediary server is used to facilitate the credential transfer, message content between sender and receiver must be opaque to an intermediary, intermediary server shall not be able to recognize the content of provisioning information or use it to provision digital credential on its own.

(Req-P2P)

(Req-P2P) If credential transfer solution supports group sharing, it shall also support limiting transfer to one device to another based on use case.

Commentary:

- This group needs to produce a spec that supports one to one sharing
- Sharing to a group is not a goal, but we want to ensure restricting sharing to one-to-one is supported

(Req-Privacy)

Transport protocol used to transfer provisioning information (e.g. secure E2E transfer protocol or intermediary server) shall prevent from correlating users between exchanges or create a social graph of users involved into transfer. Intermediary server shall not be an arbiter of identity. User identities shall not be collected, stored and used for purpose other than the credential transfer itself.

~~(Req-Privacy)~~

(Req-NonCorrelation) Transport protocol used to transfer provisioning information (e.g. secure E2E transfer protocol or intermediary server) shall prevent from correlating users between exchanges or create a social graph of users involved into transfer.

(Req-NonIdentity) Intermediary server shall not be an arbiter of identity.

~~(Req-NonCollection) User identities shall not be collected, stored and used for purpose other than the credential transfer itself.~~

(Req-SenderTrust)

In the case when an intermediary server is used to facilitate the credential transfer, sender device should establish trusted relationship with the intermediary server. Intermediary server shall be able to verify that the sender device is in good standing and content generated by the sender device can be trusted by the intermediary. The trust mechanism could be proprietary or publicly verifiable (e.g. WebAuthN). This is important because intermediary server shall have no visibility to the content of the provisioning information sent through it (Req-Opaque).

(Req-ReceiverTrust)

In the case when an intermediary server is used to facilitate the credential transfer, receiver device should be able to evaluate the trustworthiness of the intermediary based on agreed criteria.

(Req-Speed)

When both Sender and Receiver are online at the same time they should be able to quickly and efficiently transfer data.

(Req-Invitation)

The Receiver must be able to establish a connection with the Sender for the secure credential transfer using an invite that can be sent over any generic communication channel (e.g. sms, email, NFC).

Additional Considerations

- Consider removing provisioning partner entirely

Updates to Threat Model

- https://datatracker.ietf.org/doc/draft-lassey-tigress-threat-model/
- Combined threat model with Security and Privacy Goals
- Edited Security and Privacy Goals to be in line with commentary from working group
- Narrowed the threat model to focus on TIGRESS exchange
  - Previously, threat model was too large and had elements outside of TIGRESS

Threats and Mitigations - Core Protocol

Threat Description	Likelihood	Impact	Mitigations
An Attacker with physical access to the victim's phone initiates a share of a Credential to the Attacker's device	MED	HIGH	Implementers SHOULD take sufficient precautions to ensure that the device owner is in possession of the device when initiating a share such as requiring authentication at share time
Attacker intercepts or eavesdrops on sharing message	HIGH	HIGH	Solution should require an end-to-end encrypted messaging channel or otherwise specify a way to share a secret out of band
Sender mistakenly sends to the wrong Receiver	HIGH	HIGH	Implementers should ensure any initiated shares can be withdrawn or revoked at any time.
Sender device compromised	MED	HIGH	

Threats and Mitigations - Intermediary Server

Threat Description	Likelihood	Impact	Mitigations
Attacker brute forces “unguessable” location	LOW	LOW	Limited TTL of storage, rate limiting of requests
Attacker intercepts encryption key	MED	MED	Separate transmission of encryption key and unguessable location
Attacker intercepts encryption key and unguessable location	MED	HIGH	Implementor should warn users about sharing credentials to groups
Attacker compromises intermediary server	LOW	LOW	Content on the server is encrypted
Attacker uses intermediary server to store unrelated items (i.e. cat pictures)	HIGH	LOW	intermediary server should have tight size limits and TTLS to discourage misuse

1-Pager WebDAV

- https://datatracker.ietf.org/doc/draft-tigress-webdav-impl/
- Explanation of how to implement TIGRESS with WebDAV protocol
  - Original proposed solution used an intermediary with HTTP, WebDAV had natural extensions that make sense
- Implement an intermediary WebDAV server where sender and receiver can exchange messages
- Possible option: Extend WebDAV with push notifications for use cases that require multiple round trips and OpenGraph for user previews

1-Pager Signal

- https://datatracker.ietf.org/doc/draft-tigress-signal-impl/
- Implement Signal Protocol on an intermediary server
  - Would provide secure, end to end encrypted message exchange between sender and receiver
- Some considerations:
  - Intermediate Signal server has to be implemented to support Signal Protocol or user accounts have to be created within Signal Application servers.
  - Intermediate server (servers) in Signal Protocol require user identities / authentication and device identities

1-Pager GSS API

- https://datatracker.ietf.org/doc/draft-tigress-gssapi-impl/
- GSS API does not define communication channel, so would use an intermediary server
- Sender would create single use auth token, create context security token, then send token + shared secret over to receiver
- Some Considerations:
  - GSS-API also requires that each party have auth credentials before the communication occurs, which isn't a requirement for our use case.