Trust Framework Membership Validation

From IDESG Wiki

Contents

- 1 Introduction
- 2 Context
- 3 The Problem
- 4 The Solution
  - 4.1 Proposed formats for keys and identifiers
- 5 References and Coordination with other groups

Introduction

This topic covers the determination of one digital entity about the current membership of another digital entity in a Trust Framework.

This document is a list of the various methods that could be used together with a recommendation for standardization.

Context

Several Trust Frameworks exist to enable digital entities to establish the policies used by the digital entity are in conformance with established standards, regulations and commonly accepted policies.

Trust can be based on two sources, past behavior by the entity, or membership of the entity in a group whose membership has shown that they are trustworthy. The IDESG ecosystem seeks to provide trust by membership of digital entities in Trust Frameworks. This document describes how one digital entity can learn that some other digital entity is currently a member of a framework that they trust to provide and enforce policies that lead to trustworthy behavior. The value of the mark proving evidence of membership will only be valuable to the extent that all of the members show that they deserve the trust that the mark asserts. In this paper it is assumed that some other technology is used to create a trusted identifier of each digital entity.

A **Trust Framework** is defined by Kantara to be "a complete set of contracts, regulations or commitments that enable participating actors to rely on certain assertions by other actors to fulfill their information security requirements." In this document the objective is simply to allow two digital entities (actors) to satisfy each other as to the membership of the other in a mutually acceptable trust framework. While it does not address the actual user experience, it does provide proof of the authority of the digital entity to display user elements that correspond to the trust framework.

An **Identity Ecosystem (https://www.idesg.org/The-ID-Ecosystem/Overview)** is defined to be an online community of users and digital entities that are bound by a common set of technologies, processes and policies, that is, by one or more Trust Frameworks.

As general rule, the membership validation is strictly between digital entities and the user is unaware of the activity which is typically hidden from the user. Some inevitable confusion arises when a unique digital identifier is conflated with a user understandable name for a digital entity. In particular there is no standard way for the RP to acquire display name of the IdP using dynamic registration. The only methods available today require static (manual) registration of the RP with the IdP. Since the user will always be the ultimate source of trust for activities in the digital realm, it is necessary that digital identities are not created with the specific intent to confuse trust decisions. It is understood that this is not an easy requirement to satisfy.

The Problem

No widely accepted standard method exists for distinct digital entities in a trust framework to prove their conformance with the policy standards of the trust framework to each other.

No widely accepted standard method exists for securely identifying digital entities or trust frameworks with a name that a user can use for a trust deciscion. The best that exists today is TLS Extended Validation Certificates (https://cabforum.org/extended-validation-2/) which do have a short host name. The user involvement in trust decisions does need to be described in any specification of trust frameworks. In some cases it is the guardian of the user that needs to be involved in the trust decision.

The Solution

Several potential solutions were considered.

1. Public Key Infrastructure (PKI) (https://en.wikipedia.org/wiki/Public_key_infrastructure) depends on the existence of a chain of trust back to a root Trust Anchor (https://en.wikipedia.org/wiki/Trust_anchor). It would be possible for a Trust Framework to establish their own Trust Anchor and issue certificates. Creating a Certificate Authority for Trust Framework Membership would be very expensive.
2. Online Certificate Status Protocol (https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol) can be used to test the validity of a certificate without relying on the CRL of PKI. It could be the basis for trust framework membership validation. Using existing service provider certificates and providing some sort of online check of that cert with the Framework membership service would be much less expensive.
3. Account Authority Digital Signature (AADS) (http://www.garlic.com/~lynn/aadsover.htm) uses the public key technology without PKI. Ignoring the X.509 certificates of PKI completely would result in the lowest cost from Framework Membership Validation. The following solution is based on AADS.

These are some of the ways to establish trust, they all require the existence of a Trust Anchor at a well-know address of the Trust Framework. Note that TLS (HTTPS) is required before trust of digital entities can be considered.

1. Use the URL of the digital entity to identify it. While the URL was not designed to be user friendly, many security schemes depend on the user understanding the importance of the URL as a unique identifier for a digital entity. Note the confusion created by conflating a user understandable name with a unique digital identifier. This method requires one of the following to be used as the digital entity's unique identifier:
    1. The host name. This is simple, generally human readable and contains no weird punctuation. Many browser users understand the meaning of this name, but there are attacks using non-ASCII characters that confuse users with names similar to ones that the users do trust. Note the confusion created by conflating a user understandable name with a unique digital identifier.
    2. The host name plus the port number. This is the method selected by the TLS standard. This appears to users of commercial sites as the same as the above choice since commercial sites use the default port number 443.
    3. The host name plus the port number plus the path. This is the method selected by the OpenID standards. It is very easy to confuse users with this choice.
2. Use the public key of the SSL cert for the identity of the digital entity.
3. The service provider creates a name, potentially localized, by which it would like to be known.

If the service wants to use an email address as an identifier for the digital entity to enable the user to contact the manufacturer, that should be permitted.

Create a web site for the Trust Anchor that can be queried using some part of the identifier. This site should allow search terms and return all of the identifiers that match the query. This step is required for proving membership in a Trust Framework, but is optional for digital entities that just wish to self-assert their identifier and policies. In an ideal world the policies should be machine readable as well as user understandable.

Proposed formats for keys and identifiers

The Trust Anchor end point MUST respond to queries with a list of all members matching the query with a key and at least one identity. The format of the identity can be specified within the range listed below. The Trust Anchor or any service Provider would provide configuration information with a well-known URI (registered with the IETF) such as:

```
https://idesg.org/.wellknown/idesg-configuration
```

The Trust Anchor MUST have a logo and associated URL describing the category of each member and the terms and conditions that are met to permit the display of the logo. The Trust Anchor SHOULD provide the logo in different resolutions as might be needed by member web sites.

The Trust Anchor MUST provide all data attributes (claims) that are defined as mandatory by the Trust Framework. For example many trust frameworks will have levels of assurance or access that are assigned to members based on their role.

The service provider MUST provide public keys in JWK Set format, such as the following 2048-bit RSA key:

```
{
 "keys": [
   {
     "alg": "RS256",
     "e": "AQAB",
     "n": "o80vbR0ZfMhjZ...",
     "kty": "RSA",
     "kid": "rsa1"
   }
```

```
 ]
}
```

The service provider MAY provide public keys in X.509 format if the certificate template includes the identifier (DN) of the service provider and of the framework(s) that have certified it.

The service Provider MUST associate their public keys with an identity that is either:

```
The URI without the scheme (which could be either https:// or mailto://)
```

```
The X.509 CN provided that is user-friendly.
```

```
A localized name of its own choosing.
```

The service Provider MAY provide other identities, attributes and service documentation, such as URL pointers to policies and service terms of use.

References and Coordination with other groups

- Basic interactions in a single Trust Framework was an early start at building a standard set of interactions between digital entities involved in a Trust Framework.
- Federation for Identity and Cross-Credentialing Systems (FiXs) (http://fixs.org) works for standardization within the U. S. Federal government. It has no specific protocols of its own.
- Transglobal Secure Collaboration Program (http://tscp.org) was originally formed to use U. S. DoD PIV smart cards to establish trusted connections among the DoD Contractors. It has since developed a NIST pilot program (https://www.nist.gov/itl/tig/pilot-projects#summaries) to show how that trusted connection can be expanded beyond their narrow scope for use with financial institutions.
- The Trust Frameworks Catalog list In Common, Kantara and potentially other trust frameworks recognized by the IDESG.
- The JSON Web Key (JWK) IETF RFC 7515 (https://tools.ietf.org/html/rfc7517) is to be used for the key set.
- The Core OpenID Connect spec (http://openid.net/specs/openid-connect-core-1_0.html#ClaimsLanguagesAndScripts) contains information on localization of names.
- The OpenID Connect Discovery Document (https://openid.net/specs/openid-connect-discovery-1_0.html) contains other information that could be reported by the Trust Anchor to help understand the scope of their work.

```
 "service_documentation":
   "http://server.example.com/connect/service_documentation.html",
 "ui_locales_supported":
   ["en-US", "en-GB", "en-CA", "fr-FR", "fr-CA"]
```

- The mobile software registration (http://openid.net/wordpress-content/uploads/2014/04/draft-mobile-registration-01.html) profile from OpenID does show the way to provide information about the client (RP) software, but even it has not defined a friendly name that a user could understand.

Categories: Trust | User Experience
