Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2015

J. Richer, Ed.
Bespoke Engineering
L. Johansson
Swedish University Network

June 26, 2015

Vectors of Trust
draft-richer-vectors-of-trust-00

Abstract

This document defines a mechanism for describing and signaling several aspects that go into a determination of trust placed in a digital identity transaction.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 28, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document defines a mechanism for describing and signaling several aspects that go into a determination of trust placed in a digital identity transaction. Instead of communicating

1.1. Terminology

Identity Provider (IdP) A system that manages identity information and is able to assert this information across the network through an identity API.

Relying Party (RP) A system that consumes identity information from an IdP for the purposes of logging users in.

Trust Framework A document containing business rules and legal clauses that defines how different parties in an identity transaction may act.

Trustmark A verifiable attestation that a party has proved to follow the constraints of a trust framework.

Trustmark Provider A system that issues and provides verification for trustmarks.

Vector A multi-part data structure, used here for conveying information about an authentication transaction.

Vector Component One of several constituent parts that make up a vector.

2. Background and Motivation

The NIST special publication 800-63 [SP-800-63] defines a linear scale Level of Assurance (LoA) measure that combines multiple attributes about an identity transaction into a single measure of the level of trust a relying party should place on an identity transaction. Even though this definition was originally made for a specific government use cases, the LoA scale appeared to be applicable with a wide variety of authentication use cases. This has led to a proliferation of incompatible interpretations of the same scale in different trust frameworks, preventing interoperability between these frameworks in spite of their common measurement.

Since identity proofing strength increases linearly along with credential strength, the LoA scale is also too limited for describing many valid and useful forms of an identity transaction. For example, an anonymously assigned hardware token can be used in cases where the real world identity of the subject cannot be known or is verified through some out of band mechanism.

This work seeks to decompose the elements of the LoA values in a way that they can be independently communicated from an Identity Provider to a Relying Party, making comparison between trust frameworks possible.

2.1. An Identity Model

This document assumes the following (incomplete) model for identity.

The identity subject (aka user) is associated with an identity provider which acts as a trusted 3rd party on behalf of the user with

regard to a relying party by making identity assertions about the user to the relying party.

The real-world person represented by the identity subject is in possession of a (cryptographic) primary credential bound to the user by (an agent of) identity provider in such a way that the binding between the credential and the real-world user is a representation of the identity proofing process performed by the (agent of) the identity provider to verify the identity of the real-world person.

2.2. Component Architecture

The term Vectors of Trust is based on the mathematical construct of a Vector, which is defined as an item composed of multiple independent scalar values. A vector is a set of coordinates that specifies a point in a (multi-dimensional) Cartesian coordinate space. The reader is encouraged to think of a vector of trust as a point in a coordinate system, in the simples form (described below) a 3 dimensional space that is intended to be a recognizable, if somewhat elided, model of identity subject trust.

An important goal for this work is to balance the need for simplicity (particularly on the part of the relying party) with the need for expressiveness. As such, this vector construct is designed to be composable and extensible.

All components of the vector construct MUST be orthogonal in the sense that no aspect of a component overlap an aspect of another component.

The values assigned to each component of a vector is sometimes written as an ordinal number (e.g. an integer) but MUST NOT be assumed as having inherent ordinal properties when compared to the same or other components in the vector space. In other words, 1 is different from 2, but it is dangerous to assume that 2 is always "more" (better) than 1.

3. Core components

This specification defines three orthogonal components: identity proofing, credential binding, and assertion presentation. These dimensions (as described below) are intentionally elided and SHOULD be combined with other information to form trust frameworks can be used as a basis for audits of identity providers and relying parties.

This specification also defines values for each component to be used in the absence of a more specific trust framework. It is expected that trust frameworks will provide context, semantics, and mapping to

legal statutes and business rules for each value in each component. Consequently, a particular vector value can only be compared with vectors defined in the same context. The RP MUST understand and take into account the trust framework context in which a vector is being expressed in order for it to be processed securely.

It is anticipated that trust frameworks will also define additional components.

3.1. Identity Proofing

The Identity Proofing dimension defines, overall, how strongly the set of identity attributes have been verified and vetted, and how strongly they are tied to a particular credential set. In other words, this dimension describes how likely it is that a given digital identity corresponds to a particular (real-world) identity subject.

This dimension SHALL be represented by the "P" demarcator and a level value, such as "P1", "P2", etc.

3.2. Credential Management

Below we use the term "credential" to denote the credential used by the identity subject to authenticate to the identity provider.

The Credential Binding dimension defines how strongly the credential can be verified by the IdP and trusted to be presented by the party represented by a given credential. In other words, this dimension describes how likely it is that the right person is presenting the credential to the identity provider, and how easily that credential could be spoofed or stolen. This component is intended to be a general category

This dimension SHALL be represented by the "C" demarcator and a level value, such as "C1", "C2", etc. Multiple credential factors MAY be communicated simultaneously, such as when Multi-Factor Authentication is used.

3.3. Assertion Presentation

The Assertion Presentation dimension defines how well the given digital identity can be communicated across the network without information leaking to unintended parties, and without spoofing. In other words, this dimension describes how likely it is that a given digital identity asserted was actually asserted by a given identity provider for a given transaction.

This dimension SHALL be represented by the "A" demarcator and a level value, such as "A1", "A2", etc.

4. Vectors of Trust Inititial component definitions

This specification defines the following general-purpose component definitions, which MAY be used when a more specific set is unavailable. These component values are referenced in a trustmark definition

P0 No proofing is done, data is not guaranteed to be persistent across sessions

P1 Attributes are self-asserted but consistent over time, potentially pseudonymous

P2 Identity has been proofed either in person or remotely using trusted mechanisms (such as social proofing)

P3 There is a legal or contractual relationship between the identity provider and the identified party (such as signed/notarized documents, employment records)

C0 No credential is used / anonymous public service / simple session cookies (with nothing else)

C1 Shared secret such as a username and password combination

C2 Known device with trusted enrollment process

C3 Cryptographic proof of key possession using shared key

C4 Cryptographic proof of key possession using asymmetric key

C5 Sealed hardware token / trusted biometric / TPM-backed keys

A0 No protection / unsigned bearer identifier (such as a session cookie)

A1 Signed and verifiable token, passed through the browser

A2 Signed and verifiable token, passed through a back channel

A3 Token encrypted to the relying parties key and audience protected

5. Communicating Vector Values to RPs

All three of these dimensions (and others, as they are defined in extension work) MUST be combined into a single vector that can be communicated across the wire unbroken.

All vector components MUST be individually available, MUST NOT be "collapsed" into a single value without also presenting the constituent dimensions as well.

When communicating the vectors across the wire, they MUST be protected in transit and signed by the asserting authority (such as the IdP).

5.1. On the Wire Representation

The vector MUST be represented as a period-separated ('.') list of vector components, with no specific order. A vector component type MAY occur multiple times within a single vector, separated by periods, in which case it is considered an AND of the two values. In order to simplify processing by RPs, it is RECOMMENDED that trust framework definitions carefully define component values such that they are mutually exclusive or subsumptive in order to avoid this situation where possible.

Vector components MAY be omitted from a vector. No holding space is left for an omitted vector component. If a vector component is omitted, the IdP is making no claim for that category.

For example, the vector value "P1.C3.A2" translates to pseudonymous, proof of shared key, signed back-channel verified token in the context of this specification's definitions (Section 4).

Vector values MUST be communicated along side of a trustmark definition to give the components context.

5.2. In OpenID Connect

In OpenID Connect [OpenID], the IdP MUST send the vector value as a string with the "vot" (vector of trust) claim in the ID token. The trustmark (Section 7.1) that applies to this vector MUST be sent as an HTTPS URL in the "vtm" (vector trust mark) claim to provide context to the vector.

For example:

```
{
    "iss": "https://idp.example.com/",
    "sub": "jondoe1234",
    "vot": "P1.C3.A2",
    "vtm": "https://trustmark.example.org/trustmark/idp.example.com"
}
```

5.3. In SAML

In SAML a VoT vector is communicated as an AuthenticationContextClassRef, a sample definition of which might look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
     targetNamespace="urn:x-vot:P1:C3:A2"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns="urn:x-vot:P1:C3:A2"
     finalDefault="extension"
     blockDefault="substitution"
     version="2.0">
     <xs:redefine
         schemaLocation="saml-schema-authn-context-loa-profile.xsd"/>
 <xs:annotation>
    <xs:documentation>VoT vector P1.C3.A2</xs:documentation>
 </xs:annotation>
 <xs:complexType name="GoverningAgreementRefType">
    <xs:complexContent>
       <xs:restriction base="GoverningAgreementRefType">
          <xs:attribute name="governingAgreementRef"
             type="xs:anyURI"
             fixed="draft-ietf-vot-this-document-00.txt"
             use="required"/>
       </xs:restriction>
    </xs:complexContent>
 </xs:complexType>
 </xs:redefine>
</xs:schema>
```

6. Requesting Vector Values

In some identity protocols, the RP can request that particular attributes be applied to a given identity transaction.

6.1. In OpenID Connect

In OpenID Connect [OpenID], the client can request a set of acceptable VoT values with the "vtr" (vector of trust request) claim request as part of the Request Object. The value of this field is an array of JSON strings, each string identifying an acceptable set of vector components. The components are ANDed together while the individual vector strings are ORed together. Vector request values MAY omit components, indicating that any value is acceptable.

```
{
    "vtr": ["P1.C2.C3.A2", "C5.A2"]
}
```

7. Discovery and Verification

7.1. Trustmark

When an RP receives a specific vector from an IdP, it needs to make a decision to trust the vector within a specific context. A trust framework can provide such a context, allowing legal and business rules to give weight to an IdP's claims. A trustmark is a verifiable claim to conform to a specific component of a trust framework, such as a verified identity provider. The trustmark conveys the root of trustworthiness about the claims and assertions made by the IdP.

The trustmark MUST be available from an HTTPS URL by the trust framework provider. The contents of this URL are a JSON [RFC7159] document with the following fields:

sub The issuer URL of the identity provider that this trustmark pertains to. This MUST match the corresponding issuer claim in the identity token, such as the OpenID Connect "iss" field. This MUST be an HTTPS URL.

iss The issuer URL of the trustmark provider that issues this trustmark. The URL that a trustmark is fetched from MUST start with the "iss" URL in this field. This MUST be an HTTPS URL.

P Array of strings containing identity proofing values for which the identity provider has been assessed and approved

C Array of strings containing credential strength values for which the identity provider has been assessed and approved

A Array of strings containing assertion strength values for which the identity provider has been assessed and approved

For example, the following trustmark provided by the trustmark.example.org organization applies to the idp.example.org identity provider:

```
{
  "sub": "https://idp.example.org/",
  "iss": "https://trustmark.example.org/",
  "P": ["0", "1"],
  "C": ["1", "2", "3"],
  "A": ["2", "3"]
}
```

A client wishing to check the claims made by an IdP can fetch the information from the trustmark provider about what claims the IdP is allowed to make in the first place and process them accordingly.

The means by which the RP decides which trustmark providers it trusts is out of scope for this specification and is generally configured out of band.

Though most trust frameworks will provide a third-party independent verification service for components, an IdP MAY host its own trustmark. For example, a self-hosted trustmark would look like:

```
{
  "sub": "https://idp.example.org/",
  "iss": "https://idp.example.org/",
  "P": ["0", "1"],
  "C": ["1", "2", "3"],
  "A": ["2", "3"]
}
```

7.2. Discovery

The IdP MAY list all of its available trustmarks as part of its discovery document. Trustmarks are listed in the trustmarks element which contains a single JSON [RFC7159] object. The keys of this JSON object are trustmark provider issuer URLs and the values of this object are the corresponding trustmarks for this IdP.

```
{
    "trustmark": {
         "https://trustmark.example.org/": "https://trustmark.example.org/
trustmark/idp.example.org/
    }
}
```

8. Acknowledgements

The authors would like to thank the members of the Vectors of Trust mailing list in the IETF for discussion and feedback on the concept and document.

9. References

9.1. Normative References

[OpenID] Sakimura, N., Bradley, J., and M. Jones, "OpenID Connect Core 1.0", November 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

9.2. Informative References

[SP-800-63]
, , , , , , and , "Electronic Authentication Guideline", August 2013.

Appendix A. Document History

- 00

o Created initial IETF drafted based on strawman proposal discussed on VoT list.

o Split vector component definitions into their own section to allow extension and override.

o Solidified trustmark document definition.

Authors' Addresses

Justin Richer (editor)
Bespoke Engineering

Email: ietf@justin.richer.org

Leif Johansson
Swedish University Network
Thulegatan 11
Stockholm
Sweden

Email: leifj@sunet.se
URI: http://www.sunet.se