                              Vectors of Trust

Abstract

   This document defines a mechanism for describing and signaling
   several aspects of a digital identity transaction and its
   participants. These aspects are used to determine the amount of
   trust to be placed in that transaction.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc8485.

Table of Contents

1. Introduction

   Methods for measuring trust in digital identity transactions have
   historically fallen into two main categories: either all measurements
   are combined into a single scalar value or trust decisions are
   calculated locally based on a detailed set of attribute metadata.
   This document defines a method of conveying trust information that is
   more expressive than a single value but less complex than
   comprehensive attribute metadata.

   Prior to the third edition [SP-800-63-3] published in 2017, NIST
   Special Publication 800-63 [SP-800-63-2] used a single scalar
   measurement of trust called a Level of Assurance (LoA). An LoA can
   be used to compare different transactions within a system at a coarse
   level. For instance, an LoA4 transaction is generally considered
   more trusted (across all measured categories) than an LoA2
   transaction. The LoA for a given transaction is computed by the
   Identity Provider (IdP) and is consumed by a Relying Party (RP).
   Since the trust measurement is a simple numeric value, it's trivial
   for RPs to process and compare. However, since each LoA encompasses
   many different aspects of a transaction, it can't express many real-
   world situations. For instance, an anonymous user account might have
   a very strong credential, such as would be common of a whistle-blower
   or political dissident. Despite the strong credential, the lack of
   identity proofing would make any transactions conducted by the
   account to fall into a low LoA. Furthermore, different use cases and
   domains require subtly different definitions for their LoA
   categories, and one group's LoA2 is not equivalent or even comparable
   to another group's LoA2.

   Attribute-Based Access Control (ABAC) systems used by RPs may need to
   know details about a user's attributes, such as how recently the
   attribute data was verified and by whom. Attribute metadata systems
   are capable of expressing extremely fine-grained detail about the
   transaction. However, this approach requires the IdP to collect,
   store, and transmit all of this attribute data for the RP's
   consumption. The RP must process this data, which may be prohibitive
   for trivial security decisions.

   The Vectors of Trust (VoT) approach proposed in this document seeks a
   balance between these two alternatives by allowing expression of
   multiple aspects of an identity transaction (including but not
   limited to identity proofing, credential strength, credential
   management, and assertion strength), without requiring full attribute
   metadata descriptions. This method of measurement gives more
   actionable data and expressiveness than an LoA, but it is still
   relatively easy for the RP to process. It is anticipated that VoT
   can be used alongside more detailed attribute metadata systems, such

as the one proposed by NISITIR 8112 [NISTIR-8112]. The RP can use
the vector value for most basic decisions but be able to query the
IdP for additional attribute metadata where needed. Furthermore, for
RPs that do not have a need for the vector's more fine-grained
detail, it is anticipated that some trust frameworks will provide a
simple mapping between certain sets of vector values to LoAs. In
such systems, an RP is given a choice of how much detail to request
from the IdP in order to process a given transaction.

This document defines a data model for these vectors and an on-the-
wire format for conveying them between parties. The values of the
vectors defined by the data model are anchored in a trust definition.
This document also provides guidance for defining values for use in
conveying this information, including four component categories and
guidance on defining values within those categories. Additionally,
this document defines a general-purpose set of component values in an
appendix (Appendix A) for use cases that do not need something more
specific.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in
BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

1.2. Terminology

Identity Federation: A protocol in which an Identity Provider (IdP)
   asserts a user's identity information to an RP. through the use
   of a cryptographic assertion or other verifiable mechanism, or a
   system implementing such a protocol. It is also referred to
   simply as "federation".

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

Identity Subject: The individual (user) engaging in the identity
   transaction, that is, being identified by the identity provider to
   the RP.

Identity Proofing: The process of verifying and validating that a
   set of identity attributes belongs to a real-world identity
   subject.

Primary Credential: The means used by the identity subject to
   authenticate to the identity provider.

Federated Credential: The assertion presented by the IdP to the RP
   across the network to authenticate the user.

Relying Party (RP): A system that consumes identity information from
   an IdP for the purposes of authenticating the user.

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

Trustmark: A URL referencing a specific trust framework and its
   definition of vector components and vector component values.

Trustmark Provider: Defines the trust framework referenced by its
   trustmark and can verify that a given system (such as an identity
   provider) is both capable of asserting and allowed to assert the
   vector component values it is claiming.

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

Vector Component: One of several constituent parts that make up a
   vector, indicating a category of information.

Vector Component Value: One of the values applied to a vector
   component within a vector.

1.3. Identity Model

This document assumes the following model for identity based on
identity federation technologies:

The identity subject (also known as the user) is associated with an
identity provider that acts as a trusted third party on behalf of the
user, with regard to an RP by making identity assertions about the
user to the RP.

The real-world individual represented by the identity subject is in
possession of a primary credential bound to the identity subject by
the identity provider (or an agent thereof) 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
identity provider (or an agent thereof) to verify the identity of the

real-world individual. This information is carried across the
network as part of an identity assertion presented to the RP during
the authentication transaction.

1.4. Component Architecture

The term "Vectors of Trust" is inspired by the mathematical construct
of a vector, which is defined as an item composed of multiple
independent values.

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

The vector is constructed of orthogonal components, such that no
aspect of a component overlaps an aspect of another component, as
much as is possible.

2. Component Dimension Definitions

This specification defines four orthogonal components: identity
proofing, primary credential usage, primary credential management,
and assertion presentation.

This specification also defines values for each of these components
to be used in the absence of a more specific trust framework in
Appendix A. 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 context of a specific trust framework. The RP
MUST understand and take into account the trust framework context in
which a vector is being expressed in order to process a vector
correctly.

Each component is identified by a demarcator consisting of a single
uppercase ASCII letter in the range "[A-Z]". The demarcator SHOULD
reflect the category with which it is associated in a natural manner.
Demarcators for components MUST be registered as described in
Section 7. It is anticipated that trust framework definitions will
use this registry to define specialized components, but it is
RECOMMENDED that trust frameworks reuse existing components
categories wherever possible. The same demarcator MUST NOT be used
for two different dimensions, and different trust frameworks SHOULD
use the same demarcator for similar information. It is further
anticipated that there will be relatively few component dimensions

over time, and this specification defines four general-purpose
categories in this section. Note that since the processing for all
vector values is contextual to a trust framework, the exact semantics
of interpreting a component will vary based on the trust framework in
use.

The value for a given component within a vector of trust is defined
by its demarcator character followed by a single digit or lowercase
ASCII letter in the range "[0-9a-z]". Categories that have a natural
ordering SHOULD prefer digits, with larger digits indicating stronger
assertions than smaller digits. Categories that do not have a
natural ordering, or that can have an ambiguous ordering, SHOULD
prefer letters. Note that while letters could also imply order, they
can also more naturally be used mnemonically. Trust frameworks MAY
use any possible values within a category without the need for them
to be contiguous.

Categories MAY use both letters and digits simultaneously. For
example, a category could define "0" as meaning "no statement is
made" while using letters such as "a", "b", and "c" for normal values
to indicate specific options. Another system could have an ordered
base set of digits with additional details provided by letters.

Each component MAY be repeated with multiple different values within
a single vector, representing the logical AND of the values (see
Section 3.1 for details). The same component and value combination
MUST NOT be repeated within a single vector. For example, a vector
could contain both "P1" and "Pa" but not two instances of "P1". A
trust framework MAY define additional restrictions on combinations of
values.

Regardless of the type of value, the RP MUST NOT assume that the
values assigned to each component of a vector have inherent ordinal
or subsumptive properties when compared to the same or other
components in the vector space without specific knowledge of the
trust framework in use. In other words, "1" is always different from
"2", but it is dangerous to assume that "2" is always better than "1"
or that "2" satisfies all the requirements of "1".

2.1. Identity Proofing (P)

The identity proofing dimension defines, overall, how strongly the
set of identity attributes have been verified and vetted. In other
words, this dimension describes how likely it is that a given digital
identity transaction corresponds to a particular (real-world)
identity subject. For example, did the user have to provide
documentation to a trusted party to prove their legal name and
address, or were they able to self-assert such values?

This dimension uses the "P" demarcator, such as "P0", "P1", etc.
Most definitions of identity proofing will have a natural ordering,
as more or less stringent proofing can be applied to an individual
being granted an account. In such cases, it is RECOMMENDED that a
digit be used for this component and that only a single value be
allowed to be communicated in a transaction.

2.2. Primary Credential Usage (C)

The primary credential usage dimension defines how strongly the
primary credential can be verified by the IdP. In other words, how
easily that credential could be spoofed or stolen. For example, did
the user log in with a password, a biometric, a cryptographic
hardware device, or some combination of the above?

This dimension uses the "C" demarcator, such as "Ca", "Cb", etc.
Most definitions of credential usage will not have an overall natural
ordering, as there may be several equivalent classes described within
a trust framework. In such cases, it is RECOMMENDED that a letter be
used for this component and that multiple distinct credential usage
factors be allowed to be communicated simultaneously, such as when
multi-factor authentication is used.

2.3. Primary Credential Management (M)

The primary credential management dimension conveys information about
the expected lifecycle of the primary credential in use, including
its binding, rotation, and revocation. In other words, the use and
strength of policies, practices, and security controls used in
managing the credential at the IdP and its binding to the intended
individual. For example, can the user bring their own cryptographic
device or is one provided by the IdP?

This dimension uses the "M" demarcator, such as "Ma", "Mb", etc.
Most definitions of credential management will not have an overall
natural ordering, though there can be preference and comparison
between values in some circumstances. In such cases, it is
RECOMMENDED that a letter be used for this component and that
multiple distinct values be allowed to be communicated
simultaneously.

2.4. Assertion Presentation (A)

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 was asserted by a given identity provider for the

identity subject of a given transaction. While this information is
largely already known by the RP as a side effect of processing an
identity assertion in a federation protocol, this dimension is still
very useful when the RP requests a login (see Section 4) and when
describing the capabilities of an IdP. This value also allows the RP
to detect when an assertion is presented in a manner it was not
intended for, as may be the case with an attack.

This dimension uses the "A" demarcator, such as "Aa", "Ab", etc.
Most definitions of assertion presentation will not have an overall
natural ordering. In such cases, it is RECOMMENDED that a letter be
used for this component and that multiple values be allowed to be
communicated simultaneously.

3. Communicating Vector Values to RPs

A vector of trust is designed to be used in the context of an
identity and authentication transaction, providing information about
the context of a federated credential. The vector therefore needs to
be able to be communicated in the context of the federated credential
in a way that is strongly bound to the assertion representing the
federated credential.

This vector has several requirements for use.

o All applicable vector components and values need to be combined
   into a single vector.

o The vector can be communicated across the wire unbroken and
   untransformed.

o All vector components need to remain individually available, not
   "collapsed" into a single value.

o The vector needs to be protected in transit.

o The vector needs to be cryptographically bound to the assertion
   that it is describing.

o The vector needs to be interpreted in the context of a specific
   trust framework definition identified by a trustmark URL.

These requirements lead us to defining a simple string-based
representation of the vector that can be incorporated within a number
of different locations and protocols without further encoding.

3.1. On-the-Wire Representation

The vector MUST be represented as a period-separated ('.') list of
vector components. A vector component type can occur multiple times
within a single vector, but a specific value of a vector component
cannot occur more than once in a single vector. That is, while
"Cc.Cd" is a valid vector, "Cc.Cc" is not. Multiple values for a
component are considered a logical AND of the values.

Vector component values MAY appear in any order within a vector, and
the RP MUST consider different orderings of the same vector
equivalent during processing. For example, "P1.Cc.Cd.Aa",
"Aa.Cc.Cd.P1", "Cd.P1.Cc.Aa", and "Aa.P1.Cd.Cc" are all considered
equivalent to each other.

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 vector is making no claim for that component. No
default values are assumed for a missing component category.

Vector values MUST be communicated along with a trustmark URL (see
Section 5) to give the components and component values context. The
trustmark MUST be cryptographically bound to the vector value, such
as the two values being carried together in a signed assertion. A
vector value without context is unprocessable, and vectors defined in
different contexts are not directly comparable as whole values.
Different trust frameworks MAY reuse component definitions (including
their values), but processing of such cross-context values is outside
the scope of this specification.

For example, the vector "P1.Cc.Ab" translates to "pseudonymous, proof
of shared key, signed browser-passed verified assertion, and no claim
made toward credential management" in the context of this
specification's definitions (see Appendix A). A different vector
"Cb.Mc.Cd.Ac" translates to "known device, full proofing required for
credential issuance and rotation, cryptographic proof of possession
of a shared key, signed back-channel verified assertion, and no claim
made toward identity proofing" in the same context. Since no claim
is made here for identity proofing, no specific value can be assumed
by the RP. Note that this doesn't mean the user wasn't proofed at
all: it's possible that the user was fully proofed to the highest
capabilities within the trust framework, but here the IdP is not
making any specific claim about proofing to the RP, perhaps to
protect the user's privacy.

3.2. In OpenID Connect

   In OpenID Connect [OpenID], the IdP MUST send the vector as a string
   within the "vot" (vector of trust) claim in the ID token. The
   trustmark (see Section 5) that applies to this vector MUST be sent as
   a URL in the "vtm" (vector trust mark) claim to provide context to
   the vector.

   The "vot" and "vtm" claims are interpreted by the RP to apply to the
   entire identity transaction and not necessarily to any one attribute
   specifically.

   For example, assume that for the given trustmark, the body of an ID
   token claiming "pseudonymous, proof of shared key, signed back-
   channel verified token, and no claim made toward credential
   management" could look like this JSON object [RFC8259] payload of the
   ID token.

```
{
    "iss": "https://idp.example.com/",
    "sub": "jondoe1234",
    "vot": "P1.Cc.Ac",
    "vtm": "https://example.org/vot-trust-framework"
}
```

   The body of the ID token is signed and optionally encrypted using
   JSON Object Signing and Encryption (JOSE), as per the OpenID Connect
   specification. By putting the "vot" and "vtm" values inside the ID
   token, the vector and its context are strongly bound to the federated
   credential represented by the ID token.

   Vector values MAY be returned in a token introspection [RFC7662]
   response describing the ID token or access token issued during an
   OpenID Connect transaction using the same claims.

4. Requesting Vector Values

   In some identity protocols, the RP can request that particular vector
   values be used for a given identity transaction. An RP can describe
   the particular vector component values it desires the IdP assert for
   a given identity transaction by using the same syntax as defined in
   Section 3.1. Processing and fulfillment of these requests are in the
   purview of the IdP, and details are outside the scope of this
   specification.

   Future specifications MAY define alternative ways for an RP to
   request vector values from an IdP.

4.1. In OpenID Connect

In OpenID Connect [OpenID], the client MAY request a partial set of
acceptable VoT values with the "vtr" (vector of trust) claim request
as part of the request object. The value of this field is a JSON
array of strings [RFC8259], each string identifying an acceptable set
of vector components. The component values within each vector are
ANDed together while the separate vectors are ORed together. For
example, a list of vectors in the form '["P1.Cb.Cc.Ab", "Ce.Ab"]' is
stating that either the full set of "P1 AND Cb AND Cc AND Ab"
simultaneously OR the full set of "Ce AND Ab" simultaneously are
acceptable to this RP for this transaction.

Vector request values MAY omit components, indicating that any value
is acceptable for that component category, including omission of that
component in the response vector.

The mechanism by which the IdP processes the "vtr" and maps that to
the authentication transaction are out of scope of this
specification.

5. Trustmarks

A trustmark is an HTTPS URL that references a specific set of vector
values as defined by a trust framework. This URL MUST point to a
human-readable document that describes what components and values are
valid, how they are used together, and what practices the component
values represent within the trust framework. The contents of the
trustmark URL MUST be reachable by the operators or implementors of
the RP. The URL MUST be stable over time for a given trust framework
to allow RPs to process incoming vectors in a consistent fashion.
New versions of a trust framework that require different processing
rules MUST use a different trustmark URL.

For example, <https://www.rfc-editor.org/info/rfc8485> is used as the
trustmark to reference the values defined in Appendix A.

The process of a trustmark provider determining the ability of a
particular IdP to correctly assert values from a given trust
framework is outside the scope of this specification. Determining
how an RP should apply the values of a given vector to the RP's
processing is outside the scope of this specification.

6. Defining New Vector Values

Vectors of Trust is meant to be a flexible and reusable framework for
communicating authentication data between networked parties in an
identity federation protocol. However, the exact nature of the
information needed depends on the parties requiring the information
and the relationship between them. While this document does define a
usable default set of values in Appendix A, it is anticipated that
many situations will require an extension of this specification for
their own use.

Component categories such as those defined in Section 2 are intended
to be general-purpose and reusable in a variety of trust frameworks.
Extension specifications SHOULD reuse existing category definitions
where possible. Extensions MAY create additional categories where
needed by using the registry defined in Section 7. The registry
encourages reuse and discovery of existing categories across
different trust frameworks. For example, the "P" category in another
framework SHOULD be used for identity proofing and related
information.

The values of components such as those defined in Appendix A are
intended to be contextual to the defining trust document. While this
specification's component values are intended to be general-purpose
and extensions MAY reuse the values and their definitions, trust
frameworks MUST define all allowable values. As these values are
always interpreted in the context of a trustmark, these values are
not recorded in a central registry. Consequently, a P1" value from
one framework and a "P1" value from another framework could have very
different interpretations depending on their contextual trust
framework documents, even though in both cases the "P" component is
used for identity proofing in some fashion.

Trust frameworks that implement this specification SHOULD choose
either a numerical ordering or a group category approach to component
values as described in Section 2, though combinations of both types
MAY be used. Trust frameworks MUST specify whether multiple values
are allowed for each category, and while any component category is
generally allowed to have multiple distinct values, a specific
definition of a set of values in an extension MAY limit a given
component category to a single value per transaction. It is
RECOMMENDED that trust frameworks use a "0" value to indicate an
empty or null condition for a given category (for example, no
proofing being done or no authentication token being used).

All trust frameworks that extend and implement this specification
MUST be referenced by a unique trustmark URL (see Section 5) to allow
RPs to differentiate between different trust frameworks.

7. IANA Considerations

   This specification creates one registry and registers several values
   into existing registries.

7.1. Vector of Trust Components Registry

   This specification establishes the "Vectors of Trust Components"
   registry.

   Component demarcators are registered by the Specification Required
   policy documented in [RFC8126].

   Criteria that should be applied by the designated experts includes
   determining whether the proposed registration is distinct enough from
   existing entries to warrant registration, whether it is likely to be
   of general applicability, and whether the registration description is
   clear. Since all vector processing is contextual to a trust
   framework, component demarcators that do not meet these criteria can
   still be used in trust frameworks. The registry contains vector
   components that are believed to have general applicability that can
   be used as well.

   Registration requests sent to the vot@ietf.org mailing list for
   review should use an appropriate subject (e.g., "Request to register
   Vector of Trust Component name: example"). The designated expert(s)
   will provide review within a two-week period and either approve or
   deny the registration request, communicating this decision to the
   review list and IANA. Denials should include an explanation and, if
   applicable, suggestions as to how to make the request successful.
   IANA must only accept registry updates from the designated expert(s)
   and should direct all requests for registration to the vot@ietf.org
   mailing list. If the designated experts do not respond within the
   designated period, IANA should contact the IESG for guidance.

7.1.1. Registration Template

   Demarcator Symbol:
      An uppercase ASCII letter in the range [A-Z] representing this
      component (e.g., "X").

   Description:
      Brief description of the component (e.g., "Example description").

   Change Controller:
      For IETF-stream RFCs, state "IESG". For other documents, give the
      name of the responsible party.

   Specification document(s):
      Reference to the document(s) that specifies the vector component,
      preferably including a URL that can be used to retrieve a copy of
      the document(s). An indication of the relevant sections may also
      be included but is not required.

7.1.2. Initial Registry Contents

   The "Vector of Trust Components" registry contains the definitions of
   vector components and their associated demarcators.

   o Demarcator Symbol: P
   o Description: Identity proofing
   o Change Controller: IESG
   o Specification document(s): [RFC8485]

   o Demarcator Symbol: C
   o Description: Primary credential usage
   o Change Controller: IESG
   o Specification document(s): [RFC8485]

   o Demarcator Symbol: M
   o Description: Primary credential management
   o Change Controller: IESG
   o Specification document(s): [RFC8485]

   o Demarcator Symbol: A
   o Description: Assertion presentation
   o Change Controller: IESG
   o Specification document(s): [RFC8485]

7.2. Addition to the OAuth Parameters Registry

   This specification adds the following value to the "OAuth Parameters"
   registry established by [RFC6749].

   o Name: vtr
   o Description: Vector of Trust request
   o Parameter usage location: authorization request, token request
   o Change Controller: IESG
   o Reference: [RFC8485]

7.3. Additions to JWT Claims Registry

   This specification adds the following values to the "JSON Web Token
   Claims" registry established by [RFC7519].

   o Claim name: vot
   o Claim Description: Vector of Trust value
   o Change Controller: IESG
   o Reference: [RFC8485]

   o Claim name: vtm
   o Claim Description: Vector of Trust trustmark URL
   o Change Controller: IESG
   o Reference: [RFC8485]

7.4. Additions to OAuth Token Introspection Response

   This specification adds the following values to the "OAuth Token
   Introspection Response" registry established by [RFC7662].

   o Name: vot
   o Description: Vector of Trust value
   o Change Controller: IESG
   o Reference: [RFC8485]

   o Name: vtm
   o Description: Vector of Trust trustmark URL
   o Change Controller: IESG
   o Reference: [RFC8485]

8. Security Considerations

   The vector of trust value needs to be cryptographically protected in
   transit between parties, such as by using TLS as described in
   [BCP195]. The vector of trust value must be associated with a
   trustmark by the RP processing the vector. A signed OpenID Connect
   ID Token or a similarly signed assertion from another protocol would
   fulfill this requirement by carrying both the vector value and the
   trustmark URL as claims.

   The vector value is always associated with a trustmark and needs to
   be interpreted by the RP in the context of the trust framework
   defined by that trustmark. Different trust frameworks can apply
   different interpretations to the same component value, much as was
   the case with LoA. Therefore, an RP interpreting a component value
   in the wrong context could mistakenly accept or reject a request. In

order to avoid this mistake, RPs need to reject vectors that are
defined in trust frameworks that they do not understand how to
interpret properly.

The VoT framework provides a mechanism for describing and conveying
trust information. It does not define any policies for an IdP
determining which vector component values apply to a given
transaction, nor does it define any policies for applying the values
of a vector to an RP's security decision process. These policies and
associated practices are to be agreed upon by the IdP and RP, and
they should be expressed in detail in an associated human-readable
trust framework document available at the trustmark URL.

9. Privacy Considerations

By design, vector of trust values contain information about the
user's authentication and associations that can be made thereto.
Therefore, all aspects of a vector of trust contain potentially
privacy-sensitive information and must be guarded as such. Even in
the absence of specific attributes about a user, knowledge that the
user has been highly proofed or issued a strong token could provide
more information about the user than was intended. It is recommended
that IdPs send and RPs request only the information necessary for
their use case in order to prevent inadvertent information
disclosure.

10. References

10.1. Normative References

[OpenID]   Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and
            C. Mortimore, "OpenID Connect Core 1.0", November 2014,
            <http://openid.net/specs/openid-connect-core-1_0.html>.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119,
            DOI 10.17487/RFC2119, March 1997,
            <https://www.rfc-editor.org/info/rfc2119>.

[RFC6749]   Hardt, D., Ed., "The OAuth 2.0 Authorization Framework",
            RFC 6749, DOI 10.17487/RFC6749, October 2012,
            <https://www.rfc-editor.org/info/rfc6749>.

[RFC7519]   Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token
            (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015,
            <https://www.rfc-editor.org/info/rfc7519>.

   [RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection",
              RFC 7662, DOI 10.17487/RFC7662, October 2015,
              <https://www.rfc-editor.org/info/rfc7662>.

   [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
              Writing an IANA Considerations Section in RFCs", BCP 26,
              RFC 8126, DOI 10.17487/RFC8126, June 2017,
              <https://www.rfc-editor.org/info/rfc8126>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data
              Interchange Format", STD 90, RFC 8259,
              DOI 10.17487/RFC8259, December 2017,
              <https://www.rfc-editor.org/info/rfc8259>.

10.2. Informative References

   [BCP195]   Sheffer, Y., Holz, R., and P. Saint-Andre,
              "Recommendations for Secure Use of Transport Layer
              Security (TLS) and Datagram Transport Layer Security
              (DTLS)", BCP 195, RFC 7525, May 2015,
              <https://www.rfc-editor.org/info/bcp195>.

   [NISTIR-8112]
              National Institute of Standards and Technology, "A
              Proposed Schema for Evaluating Federated Attributes", NIST
              Internal Report 8112, DOI 10.6028/NIST.IR.8112, January
              2018, <https://nvlpubs.nist.gov/nistpubs/ir/2018/
              NIST.IR.8112.pdf>.

   [SP-800-63-2]
              National Institute of Standards and Technology,
              "Electronic Authentication Guideline", NIST Special
              Publication SP 800-63-2, DOI 10.6028/NIST.SP.800-63-2,
              August 2013,
              <https://dx.doi.org/10.6028/NIST.SP.800-63-2>.

   [SP-800-63-3]
              National Institute of Standards and Technology, "Digital
              Identity Guideline", NIST Special Publication SP 800-63-3,
              DOI 10.6028/NIST.SP.800-63-3, June 2017,
              <https://doi.org/10.6028/NIST.SP.800-63-3>.

Appendix A. Vectors of Trust Default Component Value Definitions

   The following general-purpose component definitions MAY be used when
   a more specific set is unavailable. This document defines a trust
   framework for these component values. The trustmark URL of this
   trust framework is <https://www.rfc-editor.org/info/rfc8485>. All
   normative requirements following in this section apply to this trust
   framework alone.

   Other trust frameworks that extend and implement this specification
   SHOULD define their own component values as described in Section 6.
   Where possible, extensions MAY reuse specific values and definitions
   as listed here, but those specific values MUST be relisted.

A.1. Identity Proofing

   The identity proofing component of this vector definition represents
   the level of scrutiny applied to the identity subject during the
   proofing process. Higher levels are largely subsumptive of lower
   levels, such that "P2" fulfills requirements for "P1", etc. Multiple
   distinct values from this category MUST NOT be used in a single
   transaction.

   P0: No proofing is done, and 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 binding relationship between the identity provider
        and the identified party (such as signed/notarized documents and
        employment records)

A.2. Primary Credential Usage

   The primary credential usage component of this vector definition
   represents distinct categories of primary credential that MAY be used
   together in a single transaction. Multiple distinct values from this
   category MAY be used in a single transaction.

   C0: No credential is used / anonymous public service

   Ca: Simple session HTTP cookies (with nothing else)

   Cb: Known device, such as those indicated through device posture or
        device management systems

   Cc: Shared secret, such as a username and password combination

   Cd: Cryptographic proof of key possession using shared key

   Ce: Cryptographic proof of key possession using asymmetric key

   Cf: Sealed hardware token / keys stored in a trusted platform module

   Cg: Locally verified biometric

A.3. Primary Credential Management

   The primary credential management component of this vector definition
   represents distinct categories of management that MAY be considered
   separately or together in a single transaction. Many trust framework
   deployments MAY use a single value for this component as a baseline
   for all transactions and thereby omit it. Multiple distinct values
   from this category MAY be used in a single transaction.

   Ma: Self-asserted primary credentials (user chooses their own
        credentials and must rotate or revoke them manually) / no
        additional verification for primary credential issuance or
        rotation

   Mb: Remote issuance and rotation / use of backup recover credentials
        (such as email verification) / deletion on user request

   Mc: Full proofing required for each issuance and rotation /
        revocation on suspicious activity

A.4. Assertion Presentation

The assertion presentation component of this vector definition
represents distinct categories of assertion that are RECOMMENDED to
be used in a subsumptive manner but MAY be used together. Multiple
distinct values from this category MAY be used in a single
transaction.

Aa: No protection / unsigned bearer identifier (such as an HTTP
     session cookie in a web browser)

Ab: Signed and verifiable assertion, passed through the user agent
     (web browser)

Ac: Signed and verifiable assertion, passed through a back channel

Ad: Assertion encrypted to the RP's key

Acknowledgements

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