Workgroup: anima Working Group
Internet-Draft:
draft-ietf-anima-voucher-delegation-02
Published: 11 July 2022
Intended Status: Standards Track
Expires: 12 January 2023
Authors: M. Richardson
Sandelman Software Works
W. Pan
Huawei Technologies

Delegated Authority for Bootstrap Voucher Artifacts

Abstract

This document describes an extension of the RFC8366 Voucher Artifact in order to support delegation of signing authority. The initial voucher pins a public identity, and that public indentity can then issue additional vouchers. This chain of authorization can support permission-less resale of devices, as well as guarding against business failure of the BRSKI Manufacturer Authorized Signing Authority (MASA).

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 https://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 12 January 2023.

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The [RFC8366] voucher artifact provides a proof from a manufacturer's authorizing signing authority (MASA) of the intended owner of a device. This is used by an onboarding Pledge device in BRSKI ([RFC8995], [I-D.ietf-anima-constrained-voucher]), and SZTP ([RFC8572]).

There are a number of criticisms of the MASA concept. They include:

* the MASA must be reachable to the Registar during the onboarding process.

* while the use of a nonceless voucher (see [RFC8366] section 4) can permit the MASA to be offline, it still requires the public key/certificate of the Registrar to be known at issuing time. The device owner is always strongly dependent on the MASA service.

* the MASA must approve all transfers of ownership, impacting the rights of the supply chain distributors to transfer ownership as they see fit.

* if the Registrar has any nonceless vouchers, then it can not change it's public key, nor can it change which certification authority it uses.

* it is not possible for a MASA to pin ownership to a Registrar by Certification Authority plus DN.

* the creator of an assembly of parts/components can speak for the entire assembly of parts in a transparent way.

1.1. Requirements for the Delegation

This voucher artifact satisfies the following requirements:

1.1.1. Device Onboarding with Disconnected or Offline MASA

A Registrar wishes to onboard devices while it is not being connected to the Internet and MASA.

1.1.2. Resale of Devices

An owner of a device wishes to resale it which has previously been onboarded to a third party without specific authorization from the manufacturer.

1.1.3. Crypto-agility for Registrar

The owner/manager of a registrar wishes to be able to replace its domain registration key. Replacing the registration key would invalidate any previously acquired (nonceless) vouchers. Any devices which have not been onboarded, or which need to be factory reset, would not trust a replacement key.

1.1.4. Transparent Assemblers/Value-Added-Resellers

An assembly may consist of a number of parts which are onboarded to a local controller during the manufacturing process. Subsequent to this, the entire assembly will be shipped to a customer who wishes to onboard all the components. The sub-components of the assembly needs to communicate with other sub-components, and so all the parts need to transparently onboarded. (This is contrasted with an assembly where the controller acts as a security gateway. Such a gateway might be a single point of failure)

Assemblies may nest quite deeply.

1.2. Overview of Proposed Solution

The MASA will issue a voucher that delegates it's signing authority for one or more devices to a specific Registrar. This is called a "delegation voucher".

This Registrar can then operate as an authorized signing authority for the manufacturer, and can subsequently issue additional vouchers binding the pledge to new Registrars.

This delegation can potentially be repeated multiple times to enable second, third, or n-th level of resale.

The delegation voucher may be stored by the pledge for storage, to be included by the pledge in subsequent bootstrap operations. The inclusion of the delegation voucher permits next Registrar with heuristics that permit it to find the delegated authorized signing authority (DASA).

The delegation voucher pins the identity of the delegated authority using a variety of different mechanisms which are covered in Section 7.

2. Terminology

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.

Delegated Authorized Signing Authority : the Delegated Authorized Signing Authority (DASA) is a service that can generate vouchers

for one or more pledges to provide bootstrap authority, which is separated and delegated from the manufacturer.

Delegation Voucher: a Delegation Voucher is an [RFC8366] format voucher that has additional fields to provide details of the entity to which authority has been delegated.

Intermediate Voucher: a voucher that is not the final voucher linking a pledge to its owner.

End Voucher: a voucher that is the final voucher linking a pledge to its owner.

3. Delegation Voucher Artifact

The following tree diagram shows the extensions to the [RFC8366] voucher.

There are a few new fields:

delegation-enable-flag: A global enable flag to the pledge that it can be delegated (true) or not (false). With default, this flag is false, which is consistent with the voucher artifact in RFC8366.

pinned-delegation-cert-authority: An subject-public-key-info for a public key of the new DASA

pinned-delegation-cert-name: A string for the rfc822Name SubjectAltName contents of the new DASA; (XXX- is it enough, should other DNs be considered?)

delegation-voucher: One or a series of Intermediate Vouchers that delegate authority to the DASA. For the latter case, the series of Intermediate Vouchers constitute a nested structure, and the most inner voucher is from the MASA, which is called terminal voucher here

intermediate-identities: A set of voucher identities being consistent with the series of Intermediate Vouchers

delegation-countdown: Number of delegations still available. If zero or omitted, then this is a terminal voucher and may not be further delegated.

In addition, the serial-number field is no longer a plain leaf, but can also be an array (See Section 3.3).

```
module: ietf-voucher-delegated

  grouping voucher-delegated-grouping:
    +-- voucher
       +-- created-on                          yang:date-and-time
       +-- expires-on?                         yang:date-and-time
       +-- assertion
       |       ianavat:voucher-assertion
       +-- serial-number                       string
       +-- idevid-issuer?                      binary
       +-- pinned-domain-cert?                 binary
       +-- domain-cert-revocation-checks?      boolean
       +-- nonce?                              binary
       +-- last-renewal-date?                  yang:date-and-time
       +-- delegation-enable-flag?             boolean
       +-- pinned-delegation-cert-authority?   binary
       +-- pinned-delegation-cert-name?        binary
       +-- delegation-voucher?                 binary
       +-- intermediate-identities?            binary
       +-- delegation-countdown?               int16
```

3.1. YANG Module

This module uses the grouping that was created in [RFC8366] to extend the definition.

```
<CODE BEGINS> file "ietf-voucher-delegated@2020-01-06.yang"

module ietf-voucher-delegated {
  yang-version 1.1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-voucher-delegated";
  prefix "delegated";

  import ietf-restconf {
    prefix rc;
    description
      "This import statement is only present to access
       the yang-data extension defined in RFC 8040.";
    reference "RFC 8040: RESTCONF Protocol";
  }

  // maybe should import from constrained-voucher instead!
  import ietf-voucher {
    prefix "v";
  }

  organization
   "IETF ANIMA Working Group";

  contact
   "WG Web:   <http://tools.ietf.org/wg/anima/>
    WG List:  <mailto:anima@ietf.org>
    Author:   Michael Richardson
              <mailto:mcr+ietf@sandelman.ca>";

  description
  "This module extends the RFC8366 voucher format to provide
   a mechanism by which the authority to issue additional vouchers
   may be delegated to another entity

   The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
   'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY',
   and 'OPTIONAL' in the module text are to be interpreted as
   described in BCP 14 RFC 2119, and RFC8174.";

  revision "2020-01-06" {
    description
     "Initial version";
    reference
     "RFC XXXX: Voucher Profile for Delegation Vouchers";
  }

  rc:yang-data voucher-delegated-artifact {
    // YANG data template for a voucher.
```

```
  uses voucher-delegated-grouping;
}

// Grouping defined for future usage
grouping voucher-delegated-grouping {
  description
    "Grouping to allow reuse/extensions in future work.";

  uses v:voucher-artifact-grouping {

    refine voucher/pinned-domain-cert {
        mandatory  false;
    }

    augment "voucher" {
      description "Base the delegated voucher
                   upon the regular one";

      leaf delegation-enable-flag {
        type boolean;
        description
          "A global enable flag to the pledge that it can be
           delegated (true) or not (false). With default,
           this flag is false, which is consistent with
           the voucher artifact in RFC8366. ";
      }

      leaf pinned-delegation-cert-authority {
        type binary;
        description
          "An subject-public-key-info for a public key of the
           certificate authority that is to be trusted to issue
           a delegation voucher to the Registrar.
           This is not used by end-vouchers, and only valid
           when delegation-enable-flag is true.";
      }

      leaf pinned-delegation-cert-name {
        type binary;
        description
          "A string for the rfc822Name SubjectAltName contents
           which will be trusted to issue delegation vouchers.
           This is not used by end-vouchers, and only valid
           when delegation-enable-flag is true.";
      }

      leaf delegation-voucher {
        type binary;
        description
          "The intermediate voucher that delegates
```

```
            authority to the entity that signs this voucher
            is to be included here, and only valid
            when delegation-enable-flag is true.";
        }

        leaf intermediate-identities {
          type binary;
          description
            "A set of identities that will be needed to
             validate the chain of vouchers, and only valid
             when delegation-enable-flag is true. MAY BE REDUNDANT";
        }

        leaf delegation-countdown {
          type int16;
          description
          "Number of delegations still available, and only valid
             when delegation-enable-flag is true. If zero
           or omitted, then this is a terminal voucher and
           may not be further delegated";
        }
      }
    }
  }
}

<CODE ENDS>
```

3.2. Bundling of The Vouchers

[RFC8995] defines a mechanism to return a single voucher to the pledge.

This protocol requires a number of additional items to be returned to the pledge for evaluation: the series of Intermediate Vouchers that leads to the DASA, and the public keys (often as certificates) of the Registrars on the Delegation Path that leads to each Authority.

3.3. Delegation of Multiple Devices

A MASA MAY delegate multiple devices to the same Registrar by putting an array of items in the "serial-number" attributes. (XXX- how to describe this in the YANG, and the detailed mechanism, are TBD)

4. Enhanced Pledge Behavior

The use of a Delegation Voucher requires changes to how the pledge evaluates the voucher that is returned to by the Registrar.

There are no significant changes to the voucher-request that is made. The pledge continues to pin the identity of the Registrar to which it is connected, providing a nonce to establish freshness.

A pledge which has previously stored a Delegation Voucher and DASA , SHOULD include it in its voucher request. This will be in the form of a certificate provided by the "previous" owner. This allows the Registrar to discover the previous authority for the pledge. As the pledge has no idea if it connecting to an entity that it previously has connected to, it needs to include this certificate anyway.

The pledge receives a voucher from the Registrar. This voucher is called the zero voucher. It will observe that the voucher is not signed with its built-in manufacturer trust anchor and it can not verify it.

The pledge will examine the voucher to look for the "delegation-voucher" and the "intermediate-identities" attributes within the voucher. A certificate from the set of intermediate-identities is expected to validate the signature on this zeroth end-entity voucher. (XXX- This attribute can be replaced by the CMS certificate chain)

The contained delegation-voucher object is to be interpreted as an (Intermediate) Voucher. This first voucher is called the first voucher, or "voucher[1]". Generically, for voucher[i], the voucher found in the delegation-voucher is called voucher[i+1].

If voucher[i] can be validated by a built-in trust anchor, then the process is done. If not, then voucher[i] is examined in a recursive process until there are no further embedded vouchers. The last voucher[n] is expected to be validated by a built-in manufacturer trust anchor.

Once the top (n-th) voucher is found, then the pinned-certificate-authority is added to the working set of trust anchors. The "pinned-certificate-name" attribute is used along with the trust anchor to validate the certificate chain provided with the (n-1)th voucher. This is repeated (unwinding the recursive processing) until the zeroth voucher has been validated.

5. Changes to Registrar Behavior

The Registrar is the component that authenticates the pledge, makes authorization decisions, and distributes vouchers. If the vouchers is delegated, then the registrar need to co-ordinate MASA and DASA.

5.1. Discovering The Most Recent Delegated Authority to Use

The pledge continues to use its manufacturer issued IDevID when performing BRSKI-style onboarding. The IDevID contains an extension, the MASA URL (see [RFC8995] section 2.3.2). The IDevID certificate is not expected to be updated when the device is resold, nor may it be practical for an intermediate owner to be able to replace the IDevID with their own. (Some devices may support having an intermediate owner replace the IDevID, in which case this section does not apply)

The Registrar needs to be informed that it should not contact a MASA using the URL in the IDevID, but rather to contact the previous owner's DASA.

This can be accomplished by local override, as described in [RFC8995] section 5.4:

Registrars MAY include a mechanism to override
the MASA URL on a manufacturer-by-manufacturer basis, and within that override it is appropriate to provide alternate anchors. This will typically used by some vendors to establish explicit (or private) trust anchors for validating their MASA that is part of a sales channel integration.

The above override needs to be established on a per-device basis. It requires per-device configuration which is very much non-autonomic.

There are two other alternatives:

1. The Manufacturer could be aware of any Delegation Vouchers that it has issued for a particular device, and when contacted by the Registrar, it could redirect the Registrar to its DASA. And the DASA may redirect the Registrar to its delegated DASA, this process is recursive to the final DASA.

2. The Pledge could provide a signed statement from the manufacturer providing the Registrar with a pointer to the DASA.

Option 1 requires that the Registrar still contact the MASA, violating most of the goals from Section 1.1.

Option 2 requires a signed artifact, and conveniently, the Delegation Voucher is exactly the item needed. The most difficult problem is that the Pledge needs to (a) store one or more Delegation Vouchers in a non-volatile storage that survives factory reset operations, (b) attach these items to the pledge's voucher-request.

The extension to the [RFC8995] voucher-request described below provides for a contained for these Delegation Vouchers.

6. Applying The Delegation Voucher to Requirements

6.1. Case 1: Resale

This case has many application scenarios.

The simplest is that a device, previously owned by one entity is sold to another entity. This would include many large home appliances (furnace, stove, refriderator) which are either sold with the home (because they are attached), or for which there is a frequent resale market. Entire systems (HVAC, physical security, elevators) in commercial buildings also fall into this category. Many of these devices exist for decades.

The initial onboarder would obtain a delegated voucher, and would keep this voucher safe. Should the device need to be resold, this voucher is provided to the new owner. This protects the first owner from situations where the manufacturer is unwilling, or goes into bankruptcy.

A creditor, such as a bank, which may take the property, including required systems as collatoral for a loan could require that a delegated voucher be obtained. A bank would find a building that needed new systems installed difficult to resale should the bank have to foreclose. It is likely that this requirement would make devices which do not come with delegated vouchers significant liabilities, and that financial institutions (banks, insurance companies) might refuse to lend in this case.

As a different example, an owner might initially start with some hosted Registrar (in the cloud perhaps, as a service). Later on, the owner wishes to bring the Registrar in-house (or just change who is providing the Registrar service). Such an activity is effectively a "resale".

It is common when a company goes bankrupt that many of it's assets (routers, switches, desktops, as well as furniture) are sold by the court. There are many resellers of digital equipment, and they typically take the devices, factory reset them, verify that they work, and then list them for resale. Such an entity would want to have a delegated voucher for each device. Whether the delegated

voucher would be obtained from the original (bankrupt) company, by the court, or directly from the manufacturer is probably a legal problem.

Further, the pledges may be resaled many times, and when onboarding, they will receive all vouchers in order with the sale chain, firstly masa vouchour, then 1st intermidate, 2nd intermidate, till to the final dealer. In this case, the pledge's authorization form a signed voucher chain.

The following illustrates a delegation voucher for a pledge: { "ietf-voucher-delegated:voucher": { "created-on": "2020-07-14T06:28:31Z", "expire-on": "2022-07-31T01:61:80Z", "assertion": "logged", "serial-number": "JADA123456789", "delegation-enable-flag": true, "pinned-delegation-cert-authority": "base64encodedvalue", "pinned-delegation-cert-name": "base64encodedvalue", "delegation-voucher": "base64encodedvalue", "intermediate-identities": "intermediateId1", "delegation-enable-flag": 1, } }

6.2. Case 2: Assembly

In some application, many pledges which come from multiple component assembled by a system integrated. They need to to be assembled together in the first sale. In this time, the owner is assembly controller, so the pledge's voucher need to include these delegation options.

In addition, there are also transparent assembly, for example rail wagon scenario. Firstly, the assembly onboards normally to get all pledges' vouchers, then this assembly acts as intermidate registrar, who "sell" these pledges to every rail wagon registrar.

7. Constraints on Pinning The Delegated Authority

TBD

8. Privacy Considerations

YYY

9. Security Considerations

9.1. Delegation Vouchers do not expire

A significant feature of the [RFC8366] voucher is that it can be short-lived, and often renewed if needed. This goes along with the arguments that renewal is better than revocation explained better in [RFC8739]. However, in order for a delegated voucher to be useful it has to have a life longer than the pessimistic expected life of the

manufacturer (MASA). This argues for the expiry time of a voucher to be rather long (decades), if not actually infinite.

[RFC8995] makes arguments for why a Pledge dos not need to have a clock that it can trust, because it can use a nonce to verify freshness of the resulting Voucher. The Delegated Voucher can not use a nonce to verify the chain of delegated vouchers presented, although it can use a nonce for the last (non-delegated) voucher.

10. IANA Considerations

This document requires the following IANA actions:

10.1. The IETF XML Registry

This document registers a URI in the "IETF XML Registry" [RFC3688]. IANA is asked to register the following:

```
URI: urn:ietf:params:xml:ns:yang:ietf-voucher-delegated
Registrant Contact: The ANIMA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.
```

10.2. YANG Module Names Registry

This document registers a YANG module in the "YANG Module Names" registry [RFC6020]. IANA is asked to register the following:

```
name:         ietf-voucher-delegated
namespace:    urn:ietf:params:xml:ns:yang:ietf-voucher-delegated
prefix:       NONE
reference:    THIS DOCUMENT
```

11. Acknowledgements

Hello.

12. Changelog

13. References

13.1. Normative References

[I-D.ietf-anima-constrained-voucher] Richardson, M., Stok, P. V. D., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (BRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-

voucher-17, 7 April 2022, <https://www.ietf.org/archive/id/draft-ietf-anima-constrained-voucher-17.txt>.

[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>.

[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>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/info/rfc8366>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/info/rfc8995>.

13.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <https://www.rfc-editor.org/info/rfc8572>.

[RFC8739] Sheffer, Y., Lopez, D., Gonzalez de Dios, O., Pastor Perales, A., and T. Fossati, "Support for Short-Term, Automatically Renewed (STAR) Certificates in the Automated Certificate Management Environment (ACME)", RFC 8739, DOI 10.17487/RFC8739, March 2020, <https://www.rfc-editor.org/info/rfc8739>.

Appendix A. Extra references

RFC Editor, please remove this section. This section lists references in the YANG. [RFC8174], [RFC8040].

Authors' Addresses

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Wei Pan
Huawei Technologies

Email: william.panwei@huawei.com