anima Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 November 2023

M. Richardson
Sandelman Software Works
W. Pan
Huawei Technologies

9 May 2023

Operational Considerations for Voucher infrastructure for BRSKI MASA
draft-richardson-anima-masa-considerations-08

Abstract

This document describes a number of operational modes that a BRSKI Manufacturer Authorized Signing Authority (MASA) may take on.

Each mode is defined, and then each mode is given a relevance within an over applicability of what kind of organization the MASA is deployed into. This document does not change any protocol mechanisms.

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 10 November 2023.

Copyright Notice

Table of Contents

1. Introduction

[RFC8995] introduces a mechanism for new devices (called pledges) to be onboarded into a network without intervention from an expert operator.

This mechanism leverages the pre-existing relationship between a device and the manufacturer that built the device. There are two aspects to this relationship: the provision of an identity for the device by the manufacturer (the IDevID), and a mechanism which convinces the device to trust the new owner (the [RFC8366] voucher).

The manufacturer, or their designate, is involved in both aspects of this process. This requires the manufacturer (or designate) to maintain on online presence.

This document offers a number of operational considerations recommendations for operating this online presence.

The first aspect is the device identity in the form of an [ieee802-1AR] certificate that is installed at manufacturing time in the device. Some of the background for the operational considerations of building this public key infrastructure is described in [I-D.irtf-t2trg-taxonomy-manufacturer-anchors].

The second aspect is the use of the Manufacturer Authorized Signing Authority (MASA), as described in [RFC8995] section 2.5.4. The device needs to have the MASA anchor built in; the exact nature of the anchor is open to a number of possibilities which are explained in this document. This document primarily deals with a number of options for architecting the security of the MASA relationship.

There are some additional considerations for a MASA that deals with constrained vouchers as described in [I-D.ietf-anima-constrained-voucher]. In particular in the COSE signed version, there may be no PKI structure included in the voucher mechanism, so cryptographic hygiene needs a different set of tradeoffs.

2. Operational Considerations for Manufacturer Authorized Signing Authority (MASA)

The manufacturer needs to make a Signing Authority available to new owners so that they may obtain [RFC8366] format vouchers to prove ownership. This section initially assumes that the manufacturer will provide this Authority internally, but subsequent sections deal with some adjustments when the authority is externally run.

The MASA is a public facing web system. It will be subject to network load from legitimate users when a network is bootstrapped for the first time. The legitimate load will be proportional to sales.

The MASA will also be subject to a malicious load.

2.1. Deflecting unwanted TLS traffic with Client Certificates

One way to deflect unwanted users from the application framework backend is to require TLS Client Certificates for all connections. As described in Section 5.5.4 of [RFC8995], the Registrar may be authenticated with a TLS Client Certificate.

This offloads much of the defense to what is typically a hardware TLS termination system. This can be effective even if the hardware is unable to do the actual validation of the TLS Client Certificate, as validation of the certificate occurs prior to any communication with the application server.

[I-D.ietf-httpbis-client-cert-field] is a critical addition to any use of TLS offload, as the certificate used needs to be communicated to the application framework for detailed authorization.

This increases the effort requires for attackers, and if they repeat the same certificate then it becomes easier to reject such attackers if a list of invalid/unwanted clients is cached.

The use of a client certificate forces attackers to generate new key pairs and certificates for each attack.

2.2. Web framework architecture

Web framework three-tier mechanisms are a very common architecture. See [threetier] for an overview. There are Internet scale frameworks exist for Ruby (RubyOnRails), Python (Django), Java (J2EE), GO, PHP and others. The methods of deploying them and dealing with expected scale are common in most enterprise IT departments.

Consideration should be made to deploying the presentation layer into multiple data centers in order to provide resiliency against distributed denial of service (DDoS) attacks that affect all tenants of that data center.

Consideration should also be given to the use of a cloud front end to mitigate attacks, however, such a system needs to be able to securely transmit the TLS Client Certificates, if the MASA wants to identify Registrars at the TLS connection time.

The middle (application) tier needs to be scalable, but it is unlikely that it needs to scale very much on a per-minute or even per-hour basis. It is probably easier and more reliable to have application tiers do database operations across the Internet or via VPN to a single location database cluster than it is to handle asynchronous database operations resulting from geographically dispersed multi-master database systems.

But, these are local design decisions which web deployment make on a regular basis. The MASA functionality is not different than other public facing systems.

The database tables that the MASA uses scale linearly with the number of products sold, but as they are mostly read-only, they could be easily replicated in a read-only manner from a sales database.

Direct integration with a sales system could be considered, but would involve a more significant security impact analysis, so a process where the sales data is extracted to a less sensitive system is RECOMMENDED.

In any case, the manufacturer SHOULD plan for a situation where the manufacturer is no longer able or interested in running the Authority: this does not have to an unhappy situation! While the case of the manufacturer going out of business is discussed in Section 5, there are more happy events which should be prepared for. For instance, if a manufacturer goes through a merge or acquisition and it makes sense to consolidate the Signing Authority in another part of the organization.

Business continuity plan should include backing up the voucher signing keys. This may involve multiple Hardware Security Modules, and secret splitting mechanisms SHOULD be employed. For large value items, customers are going to need to review the plan as part of their contingency audits. The document [I-D.irtf-t2trg-taxonomy-manufacturer-anchors] can provide some common basis for this kind of evaluation.

The trust anchors needs to validate [RFC8366] vouchers will typically be part of the firmware loaded inot the devie firmware.

There are many models to manage these trust anchors, but in order having only a single key, a PKI infrastructure is appropriate, but not required.

On constrained devices without code space to parse and validate a public key certificate chain require different considerations, a single key may be necessary. This document does not (yet) provide appropriate considerations for that case.

What follows are a number of ways to construct a resilient PKI to sign vouchers.

2.3. Self-contained multi-product MASA, no PKI

The simplest situation is to create a self-signed End Entity certificate. That is, a public/private key pair. The certificate/ public key is embedded in the products to validate vouchers, and the private part is kept online to sign vouchers.

This situation has very low security against theft of a key from the MASA. Such a theft would result in recall of all products that have not yet been onboarded. It is very simple to operate.

2.4. Self-contained multi-product MASA, with one-level PKI

A simple way is to create an new offline certification authority (CA), have it periodically sign a new End-Entity (EE) identity's certificate. This End-Entity identity has a private key kept online, and it uses that to sign voucher requests. Note that the entity used to sign [RFC8366] format vouchers does not need to be a certificate authority.

If the public key of this offline CA is then built-in to the firmware of the device, then the devices do not need any further anchors.

There is no requirement for this CA to be signed by any other certification authority. That is, it may be a root CA. There is also no prohibition against it.

If this offline CA signs any other certificates, then it is important that the device know which End-Entity certificates may sign vouchers. This is an authorization step, and it may be accomplished it a number of ways:

1. the Distinguished Name (DN) of the appropriate End-Entity certificate can be built-in to the firmware

2. a particular policy OID may be included in certificates intended to sign vouchers

A voucher created for one product could be used to sign a voucher for another product. This situation is also mitigated by never repeating serialNumbers across product lines.

An End-Entity certificate used to sign the voucher is included in the certificate set in the CMS structure that is used to sign the voucher. The root CA's trust anchor should _also_ be included, even though it is self-signed, as this permits auditing elements in a Registrar to validate the End-Entity Certificate.

The inclusion of the full chain also supports a Trust-on-First-Use (TOFU) workflow for the manager of the Registrar: they can see the trust anchor chain and can compare a fingerprint displayed on their screen with one that could be included in packaging or other sales channel information.

When building the MASA public key into a device, only the public key contents matter, not the structure of the self-signed certificate itself. Using only the public key enables a MASA architecture to evolve from a single self-contained system into a more complex architecture later on.

2.5. Self-contained per-product MASA

A simple enhancement to the previous scenario is to have a unique MASA offline key for each product line. This has a few advantages:

* if the private keys are kept separately (under different encryption keys), then compromise of a single product lines MASA does not compromise all products.

* if a product line is sold to another entity, or if it has to go through an escrow process due to the product going out of production, then the process affects only a single product line.

* it is safe to have serialNumber duplicated among different product lines since a voucher for one product line would not validate on another product line.

The disadvantage is that it requires a private key to be stored per product line, and most large OEMs have many dozens of product lines. If the keys are stored in a single Hardware Security Module (HSM), with the access to it split across the same parties, then some of the cryptographic advantages of different private keys will go away, as a compromise of one key likely compromises them all. Given a HSM, the most likely way a key is compromised is by an attacker getting authorization on the HSM through theft or coercion.

The use of per-product MASA signing keys is encouraged.

2.6. Per-product MASA keys intertwined with IDevID PKI

The IDevID certificate chain (the intermediate CA and root CA that signed the IDevID certificate) should be included in the device firmware so that they can be communicated during the BRSKI-EST exchange.

Since they are already present, could they be used as the MASA trust anchor as well?

In order to do this there is an attack that needs to mitigated. Since the root-CA that creates IDevIDs and the root-CA that creates vouchers are the same, when validating a voucher, a pledge needs to make sure that it is signed by a key authorized to sign vouchers. In

other scenarios any key signed by the voucher-signing-root-CA would be valid, but in this scenario that would also include any IDevID, such as would be installed in any other device. Without an additional signal as to which keys can sign vouchers, and which keys are just IDevID keys, then it would be possible to sign vouchers with any IDevID private key, rather than just the designated voucher-signing key. An attacker that could extract a private key from even one instance of a product, could use that to sign vouchers, and impersonate the MASA.

The challenge with combining it into the IDevID PKI is making sure that only an authorized entity can sign the vouchers. The solution is that it can not be the same intermediate CA that is used to sign the IDevID, since that CA should have the authority to sign vouchers.

The PKI root CA therefore needs to sign an intermediate CA, or End-Entity certificate with an extension OID that is specific for Voucher Authorization. This is easy to do as policy OIDs can be created from Private Enterprise Numbers. There is no need for standardization, as the entity doing the signing is also creating the verification code. If the entire PKI operation was outsource, then there would be a benefit for standardization.

2.7. Rotating MASA authorization keys

As a variation of the scenario described in Section 2.5, there could be multiple Signing Authority keys per product line. They could be rotated though in some deterministic order. For instance, serial numbers ending in 0 would have MASA key 0 embedded in them at manufacturing time. The asset database would have to know which key that corresponded to, and it would have to produce vouchers using that key.

There are significant downsides to this mechanism:

* all of the MASA signing keys need to be online and available in order to respond to any voucher request

* it is necessary to keep track of which device trust which key in the asset database

There is no obvious advantage to doing this if all the MASA signing private keys are kept in the same device, under control of the same managers. But if the keys are spread out to multiple locations and are under control of different people, then there may be some advantage. A single MASA signing authority key compromise does not cause a recall of all devices, but only the portion that had that key embedded in it.

The relationship between signing key and device could be temporal: all devices made on Tuesday could have the same key, there could be hundreds of keys, each one used only for a few hundred devices. There are many variations possible.

The major advantage comes with the COSE signed constrained-vouchers described in [I-D.ietf-anima-constrained-voucher]. In this context, where there isn't space in the voucher for a certificate chain, nor is there code in the device to validate a certificate chain, a raw public key can sign the voucher. The (public) key used to sign is embedded directly in the firmware of each device without the benefit of any public key infrastructure, which would allow indirection of the key.

3. Operational Considerations for Constrained MASA

TBD

4. Operational Considerations for creating Nonceless vouchers

TBD

5. Business Continuity and Escow Considerations

A number of jurisdictions have legal requirements for businesses to have contingency plans in order to continue operating after an incident or disaster. Specifications include [iso22301_2019], but the problem of continuity goes back over 40 years.

The [holman2012] document defined an eight tier process to understand how data would be backed up. Tier 0 is "no off-site data", and would be inappropriate for the MASA's signing key. The question as to how much delay (downtime) is tolerable during a disaster for activating new devices. The consideration should depend upon the type of the device, and what kind of disasters are being planned for. Given current technologies for replicating databases online, a tier-4 ("Point-in-time copies") or better solution may be quite economically deployed.

A key aspect of the MASA is that it was designed as a component that can be outsourced to a third party, and this third party can leverage economies of scale to provide more resilient systems at much lower costs.

The PKI components that are used to provision the IDevID certificiates into new devices need to be operational only when the factory that produces the devices is active. The business continuity planning needs to include provision for backing up the private keys

used within the PKI. It may be enough to backup just the root CA key: the rest of the levels of the PKI can be regenerated in another location if necessary.

6. Privacy Considerations

YYY

7. Security Considerations

ZZZ

8. IANA Considerations

This document makes no IANA requests.

9. Acknowledgements

Hello.

10. Changelog

11. References

11.1. Normative References

[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>.

[I-D.ietf-anima-constrained-voucher] Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (BRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-20, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-20>.

[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>.

[I-D.richardson-anima-registrar-considerations]
Richardson, M. and W. Pan, "Operational Considerations for BRSKI Registrar", Work in Progress, Internet-Draft, draft-richardson-anima-registrar-considerations-06, 7 November 2022, <https://datatracker.ietf.org/doc/html/draft-richardson-anima-registrar-considerations-06>.

[I-D.moskowitz-ecdsa-pki]
Moskowitz, R., Birkholz, H., Xia, L., and M. Richardson, "Guide for building an ECC pki", Work in Progress, Internet-Draft, draft-moskowitz-ecdsa-pki-10, 31 January 2021, <https://datatracker.ietf.org/doc/html/draft-moskowitz-ecdsa-pki-10>.

[threetier]
Wikipedia, "Multitier architecture", December 2019, <https://en.wikipedia.org/wiki/Multitier_architecture>.

[ieee802-1AR]
IEEE Standard, "IEEE 802.1AR Secure Device Identifier", 2009, <http://standards.ieee.org/findstds/standard/802.1AR-2009.html>.

[I-D.irtf-t2trg-taxonomy-manufacturer-anchors]
Richardson, M., "A Taxonomy of operational security considerations for manufacturer installed keys and Trust Anchors", Work in Progress, Internet-Draft, draft-irtf-t2trg-taxonomy-manufacturer-anchors-00, 22 January 2023, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-taxonomy-manufacturer-anchors-00>.

[I-D.ietf-httpbis-client-cert-field]
Campbell, B. and M. Bishop, "Client-Cert HTTP Header Field", Work in Progress, Internet-Draft, draft-ietf-httpbis-client-cert-field-06, 17 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-client-cert-field-06>.

11.2. Informative References

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[BedOfNails]
Wikipedia, "In-circuit test", 2019, <https://en.wikipedia.org/wiki/In-circuit_test#Bed_of_nails_tester>.

[RambusCryptoManager]
Qualcomm press release, "Qualcomm Licenses Rambus CryptoManager Key and Feature Management Security Solution", 2014, <https://www.rambus.com/qualcomm-licenses-rambus-cryptomanager-key-and-feature-management-security-solution/>.

[kskceremony]
Verisign, "DNSSEC Practice Statement for the Root Zone ZSK Operator", 2017, <https://www.iana.org/dnssec/dps/zsk-operator/dps-zsk-operator-v2.0.pdf>.

[rootkeyceremony]
Community, "Root Key Ceremony, Cryptography Wiki", April 2020, <https://cryptography.fandom.com/wiki/Root_Key_Ceremony>.

[keyceremony2]
Digi-Sign, "SAS 70 Key Ceremony", April 2020, <http://www.digi-sign.com/compliance/key%20ceremony/index>.

[nistsp800-57]
NIST, "SP 800-57 Part 1 Rev. 4 Recommendation for Key Management, Part 1: General", 1 January 2016, <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final>.

[iso22301_2019]
ISO, "ISO 22301: Societal security Business continuity management systems Requirements", 1 January 2019, <https://www.iso.org/standard/75106.html>.

[holman2012]
Holman, E., "A Business Continuity Solution Selection Methodology", 13 March 2012, <https://share.confex.com/share/118/webprogram/Handout/Session10387/Session%2010387%20Business%20Continuity%20Solution%20Selection%20Methodology%2003-7-2012.pdf>.

Authors' Addresses

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca

Wei Pan
Huawei Technologies
Email: william.panwei@huawei.com