Stream:	Internet Engineering Task Force (IETF)
RFC:	8657
Category:	Standards Track
Published:	November 2019
ISSN:	2070-1721
Author:	H. Landau

RFC 8657
Certification Authority Authorization (CAA) Record Extensions for Account URI and Automatic Certificate Management Environment (ACME) Method Binding

Abstract

The Certification Authority Authorization (CAA) DNS record allows a domain to communicate an issuance policy to Certification Authorities (CAs) but only allows a domain to define a policy with CA-level granularity. However, the CAA specification (RFC 8659) also provides facilities for an extension to admit a more granular, CA-specific policy. This specification defines two such parameters: one allowing specific accounts of a CA to be identified by URIs and one allowing specific methods of domain control validation as defined by the Automatic Certificate Management Environment (ACME) protocol to be required.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This specification defines two parameters for the "issue" and "issuewild" Properties of the Certification Authority Authorization (CAA) DNS resource record [RFC8659]. The first, "accounturi", allows authorization conferred by a CAA policy to be restricted to specific accounts of a Certification Authority (CA), which are identified by URIs. The second, "validationmethods", allows the set of validation methods supported by a CA to validate domain control to be limited to a subset of the full set of methods that it supports.

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.

3. Extensions to the CAA Record: The "accounturi" Parameter

This document defines the "accounturi" CAA parameter for the "issue" and "issuewild" Properties defined by [RFC8659]. The value of this parameter, if specified, **MUST** be a URI [RFC3986] identifying a specific CA account.

"CA account" means an object that is maintained by a specific CA, that may request the issuance of certificates, and that represents a specific entity or group of related entities.

The presence of this parameter constrains the Property to which it is attached. Where a CAA Property has an "accounturi" parameter, a CA **MUST** only consider that Property to authorize issuance in the context of a given certificate issuance request if the CA recognizes the URI specified in the value portion of that parameter as identifying the account making that request.

A Property without an "accounturi" parameter matches any account. A Property with an invalid or unrecognized "accounturi" parameter is unsatisfiable. A Property with multiple "accounturi" parameters is unsatisfiable.

The presence of an "accounturi" parameter does not replace or supersede the need to validate the domain name specified in an "issue" or "issuewild" record in the manner described in the CAA specification [RFC8659]. CAs **MUST** still perform such validation. For example, a CAA "issue" Property that specifies a domain name belonging to CA A and an "accounturi" parameter identifying an account at CA B is unsatisfiable.

3.1. Use with ACME

An Automatic Certificate Management Environment (ACME) [RFC8555] account object **MAY** be identified by setting the "accounturi" parameter to the URI of the ACME account object.

Implementations of this specification that also implement ACME **MUST** recognize such URIs.

3.2. Use without ACME

The "accounturi" specification provides a general mechanism to identify entities that may request certificate issuance via URIs. The use of specific kinds of URIs may be specified in future RFCs, and CAs not implementing ACME **MAY** assign and recognize their own URIs arbitrarily.

4. Extensions to the CAA Record: The "validationmethods" Parameter

This document also defines the "validationmethods" CAA parameter for the "issue" and "issuewild" Properties. The value of this parameter, if specified, **MUST** be a comma-separated string of zero or more validation method labels.

A validation method label identifies a validation method. A validation method is a particular way in which a CA can validate control over a domain.

The presence of this parameter constrains the Property to which it is attached. A CA **MUST** only consider a Property with the "validationmethods" parameter to authorize issuance where the validation method being used is identified by one of the validation method labels listed in the comma-separated list.

Each validation method label **MUST** be either the label of a method defined in the "ACME Validation Methods" IANA registry [RFC8555] or a CA-specific non-ACME validation method label as defined below.

Where a CA supports both the "validationmethods" parameter and one or more non-ACME validation methods, it **MUST** assign labels to those methods. If appropriate non-ACME labels are not present in the "ACME Validation Methods" IANA registry, the CA **MUST** use labels beginning with the string "ca-", which are defined to have CA-specific meaning.

The value of the "validationmethods" parameter **MUST** comply with the following ABNF [RFC5234]:

```
   value = [*(label ",") label]
   label = 1*(ALPHA / DIGIT / "-")
```

5. Security Considerations

This specification describes an extension to the CAA record specification, increasing the granularity at which a CAA policy can be expressed. This allows the set of entities capable of successfully requesting issuance of certificates for a given domain to be restricted beyond the set of entities would otherwise be possible, while still allowing issuance for specific accounts of a CA. This improves the security of issuance for domains that choose to employ it, when combined with a CA that implements this specification.

5.1. Limited to CAs Processing CAA Records

All of the security considerations listed in [RFC8659] are inherited by this document. This specification merely enables a domain with an existing relationship with a CA to further constrain that CA in its issuance practices, where that CA implements this specification. In particular, it provides no additional security above that provided by using the unextended CAA specification alone as concerns matters relating to any other CA. The capacity of any other CA to issue certificates for the given domain is completely unchanged.

As such, a domain that, via CAA records, authorizes only CAs adopting this specification and that constrains its policy by means of this specification, remains vulnerable to unauthorized issuance by CAs that do not honor CAA records or that honor them only on an advisory basis. Where a domain uses DNSSEC, it also remains vulnerable to CAs that honor CAA records but that do not validate CAA records by means of a trusted DNSSEC-validating resolver.

5.2. Restrictions Ineffective without CA Recognition

Because the parameters of "issue" or "issuewild" CAA Properties constitute a CA-specific namespace, the CA identified by an "issue" or "issuewild" Property decides what parameters to recognize and their semantics. Accordingly, the CAA parameters defined in this specification rely on their being recognized by the CA named by an "issue" or "issuewild" CAA Property and are not an effective means of control over issuance unless a CA's support for the parameters is established beforehand.

CAs that implement this specification **SHOULD** make available documentation indicating as such, including explicit statements as to which parameters are supported. Domains configuring CAA records for a CA **MUST NOT** assume that the restrictions implied by the "accounturi" and "validationmethods" parameters are effective in the absence of explicit indication as such from that CA.

CAs **SHOULD** also document whether they implement DNSSEC validation for DNS lookups done for validation purposes, as this affects the security of the "accounturi" and "validationmethods" parameters.

5.3. Mandatory Consistency in CA Recognition

A CA **MUST** ensure that its support for the "accounturi" and "validationmethods" parameters is fully consistent for a given domain name that a CA recognizes as identifying itself in a CAA "issue" or "issuewild" Property. If a CA has multiple issuance systems (for example, an ACME-based issuance system and a non-ACME-based issuance system, or two different issuance systems resulting from a corporate merger), it **MUST** ensure that all issuance systems recognize the same parameters.

A CA that is unable to do this **MAY** still implement the parameters by splitting the CA into two domain names for the purposes of CAA processing. For example, a CA "example.com" with an ACME-based issuance system and a non-ACME-based issuance system could recognize only "acme.example.com" for the former and "example.com" for the latter, and then implement support for the "accounturi" and "validationmethods" parameters for "acme.example.com" only.

A CA that is unable to ensure consistent processing of the "accounturi" parameter or the "validationmethods" parameter for a given CA domain name as specifiable in CAA "issue" or "issuewild" Properties **MUST NOT** implement support for these parameters. Failure to do so would result in an implementation of these parameters that does not provide effective security.

5.4. URI Ambiguity

Suppose that CA A recognizes "a.example.com" as identifying itself and CA B is a subsidiary of CA A that recognizes both "a.example.com" and "b.example.com" as identifying itself.

Suppose that both CA A and CA B issue account URIs of the form:

```
   "urn:example:account-id:1234"
```

If the CA domain name in a CAA record is specified as "a.example.com", then this could be construed as identifying account number 1234 at CA A or at CA B. These may be different accounts, creating ambiguity.

Thus, CAs **MUST** ensure that the URIs they recognize as pertaining to a specific account of that CA are unique within the scope of all domain names that they recognize as identifying that CA for the purpose of CAA record validation.

CAs **SHOULD** satisfy this requirement by using URIs that include an authority (see Section 3.2 of [RFC3986]):

```
   "https://a.example.com/account/1234"
```

5.5. Authorization Freshness

The CAA specification [RFC8659] governs the act of issuance by a CA. In some cases, a CA may establish authorization for an account to request certificate issuance for a specific domain separately from the act of issuance itself. Such authorization may occur substantially prior to a certificate issuance request. The CAA policy expressed by a domain may have changed in the meantime, creating the risk that a CA will issue certificates in a manner inconsistent with the presently published CAA policy.

CAs **SHOULD** adopt practices to reduce the risk of such circumstances. Possible countermeasures include issuing authorizations with very limited validity periods, such as an hour, or revalidating the CAA policy for a domain at certificate issuance time.

5.6. Use with and without DNSSEC

The "domain validation" model of validation commonly used for certificate issuance cannot ordinarily protect against adversaries who can conduct global man-in-the-middle attacks against a particular domain. A global man-in-the-middle attack is an attack that can intercept traffic to or from a given domain, regardless of the origin or destination of that traffic. Such an adversary can intercept all validation traffic initiated by a CA and thus appear to have control of the given domain.

Where a domain is signed using DNSSEC, the authenticity of its DNS data can be assured, providing that a given CA makes all DNS resolutions via a trusted DNSSEC-validating resolver. A domain can use this Property to protect itself from the threat posed by an adversary capable of performing a global man-in-the-middle attack against that domain.

In order to facilitate this, a CA validation process must either rely solely on information obtained via DNSSEC or meaningfully bind the other parts of the validation transaction using material obtained via DNSSEC.

The CAA parameters described in this specification can be used to ensure that only validation methods meeting these criteria are used. In particular, a domain secured via DNSSEC **SHOULD** either:

1. Use the "accounturi" parameter to ensure that only accounts that it controls are authorized to obtain certificates, or
2. Exclusively use validation methods that rely solely on information obtained via DNSSEC and use the "validationmethods" parameter to ensure that only such methods are used.

A CA supporting the "accounturi" parameter or the "validationmethods" parameter **MUST** perform CAA validation using a trusted DNSSEC-validating resolver.

"Trusted" in this context means that the CA both trusts the resolver itself and ensures that the communications path between the resolver and the system performing CAA validation is secure. It is **RECOMMENDED** that a CA ensure this by using a DNSSEC-validating resolver running on the same machine as the system performing CAA validation.

The use of the "accounturi" parameter or the "validationmethods" parameter does not confer additional security against an attacker capable of performing a man-in-the-middle attack against all validation attempts made by a given CA that is authorized by CAA where:

1. A domain does not secure its nameservers using DNSSEC, or
2. That CA does not perform CAA validation using a trusted DNSSEC-validating resolver.

Moreover, the use of the "accounturi" parameter or the "validationmethods" parameter does not mitigate man-in-the-middle attacks against CAs that do not validate CAA records or that do not do so using a trusted DNSSEC-validating resolver, regardless of whether or not those CAs are authorized by CAA; see Section 5.1.

In these cases, the "accounturi" and "validationmethods" parameters still provide an effective means of administrative control over issuance, except where control over DNS is subdelegated (see below).

5.7. Restrictions Supersedable by DNS Delegation

CAA records are located during validation by walking up the DNS hierarchy until one or more records are found. CAA records are therefore not an effective way of restricting or controlling issuance for subdomains of a domain, where control over those subdomains is delegated to another party (such as via DNS delegation or by providing limited access to manage subdomain DNS records).

5.8. Misconfiguration Hazards

Because the "accounturi" and "validationmethods" parameters express restrictive security policies, misconfiguration of said parameters may result in legitimate issuance requests being refused.

5.9. Revelation of Account URIs

Because CAA records are publicly accessible, the use of the "accounturi" parameter enables third parties to observe the authorized account URIs for a domain. This may allow third parties to identify a correlation between domains if those domains use the same account URIs.

CAs are encouraged to select and process account URIs under the assumption that untrusted third parties may learn of them.

6. IANA Considerations

This document has no IANA actions. As per [RFC8659], the parameter namespace for the CAA "issue" and "issuewild" Properties has CA-defined semantics, and the identifiers within that namespace may be freely and arbitrarily assigned by a CA. This document merely specifies recommended semantics for parameters of the names "accounturi" and "validationmethods", which CAs may choose to adopt.

7. Normative References

[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>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8659] Hallam-Baker, P., Stradling, R., and J. Hoffman-Andrews, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 8659, DOI 10.17487/RFC8659, November 2019, <https://www.rfc-editor.org/info/rfc8659>.

Appendix A. Examples

The following shows an example DNS zone file fragment that nominates two account URIs as authorized to issue certificates for the domain "example.com". Issuance is restricted to the CA "example.net".

```
example.com. IN CAA 0 issue "example.net; \
  accounturi=https://example.net/account/1234"
example.com. IN CAA 0 issue "example.net; \
  accounturi=https://example.net/account/2345"
```

The following shows a zone file fragment that restricts the ACME methods that can be used; only ACME methods "dns-01" and "xyz-01" can be used.

```
example.com. IN CAA 0 issue "example.net; \
  validationmethods=dns-01,xyz-01"
```

The following shows an equivalent way of expressing the same restriction:

```
example.com. IN CAA 0 issue "example.net; validationmethods=dns-01"
example.com. IN CAA 0 issue "example.net; validationmethods=xyz-01"
```

The following shows a zone file fragment in which one account can be used to issue with the "dns-01" method and one account can be used to issue with the "http-01" method.

```
example.com. IN CAA 0 issue "example.net; \
  accounturi=https://example.net/account/1234; \
  validationmethods=dns-01"
example.com. IN CAA 0 issue "example.net; \
  accounturi=https://example.net/account/2345; \
  validationmethods=http-01"
```

The following shows a zone file fragment in which only ACME method "dns-01" or a CA-specific method "ca-foo" can be used.

```
example.com. IN CAA 0 issue "example.net; \
  validationmethods=dns-01,ca-foo"
```

Author's Address

Hugo Landau
Email: hlandau@devever.net