

Network Working Group
Internet-Draft
Obsoletes: [6844](#) (if approved)
Intended status: Standards Track
Expires: August 8, 2019

P. Hallam-Baker

R. Stradling
Sectigo
J. Hoffman-Andrews
Let's Encrypt
February 04, 2019

**DNS Certification Authority Authorization (CAA) Resource Record
draft-ietf-lamps-rfc6844bis-05**

Abstract

The Certification Authority Authorization (CAA) DNS Resource Record allows a DNS domain name holder to specify one or more Certification Authorities (CAs) authorized to issue certificates for that domain name. CAA Resource Records allow a public Certification Authority to implement additional controls to reduce the risk of unintended certificate mis-issue. This document defines the syntax of the CAA record and rules for processing CAA records by certificate issuers.

This document obsoletes [RFC 6844](#).

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 August 8, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Definitions	3
2.1.	Requirements Language	3
2.2.	Defined Terms	4
3.	Relevant Resource Record Set	5
4.	Mechanism	6
4.1.	Syntax	6
4.1.1.	Canonical Presentation Format	7
4.2.	CAA issue Property	8
4.3.	CAA issuewild Property	9
4.4.	CAA iodef Property	10
4.5.	Critical Flag	11
5.	Security Considerations	11
5.1.	Use of DNS Security	12
5.2.	Non-Compliance by Certification Authority	12
5.3.	Mis-Issue by Authorized Certification Authority	12
5.4.	Suppression or Spoofing of CAA Records	12
5.5.	Denial of Service	13
5.6.	Abuse of the Critical Flag	13
6.	Deployment Considerations	13
6.1.	Blocked Queries or Responses	14
6.2.	Rejected Queries and Malformed Responses	14
6.3.	Delegation to Private Nameservers	14
6.4.	Bogus DNSSEC Responses	14
7.	Differences versus RFC6844	15
8.	IANA Considerations	15
9.	Acknowledgements	16
10.	References	16
10.1.	Normative References	16
10.2.	Informative References	17
	Authors' Addresses	17

[1.](#) Introduction

The Certification Authority Authorization (CAA) DNS Resource Record allows a DNS domain name holder to specify the Certification Authorities (CAs) authorized to issue certificates for that domain name. Publication of CAA Resource Records allows a public

Certification Authority to implement additional controls to reduce the risk of unintended certificate mis-issue.

Like the TLSA record defined in DNS-Based Authentication of Named Entities (DANE) [[RFC6698](#)], CAA records are used as a part of a mechanism for checking PKIX [[RFC6698](#)] certificate data. The distinction between the two specifications is that CAA records specify an authorization control to be performed by a certificate issuer before issue of a certificate and TLSA records specify a verification control to be performed by a relying party after the certificate is issued.

Conformance with a published CAA record is a necessary but not sufficient condition for issuance of a certificate.

Criteria for inclusion of embedded trust anchor certificates in applications are outside the scope of this document. Typically, such criteria require the CA to publish a Certification Practices Statement (CPS) that specifies how the requirements of the Certificate Policy (CP) are achieved. It is also common for a CA to engage an independent third-party auditor to prepare an annual audit statement of its performance against its CPS.

A set of CAA records describes only current grants of authority to issue certificates for the corresponding DNS domain name. Since certificates are valid for a period of time, it is possible that a certificate that is not conformant with the CAA records currently published was conformant with the CAA records published at the time that the certificate was issued. Relying parties **MUST NOT** use CAA records as part of certificate validation.

CAA records **MAY** be used by Certificate Evaluators as a possible indicator of a security policy violation. Such use **SHOULD** take account of the possibility that published CAA records changed between the time a certificate was issued and the time at which the certificate was observed by the Certificate Evaluator.

[2.](#) Definitions

[2.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC8174](#)].

2.2. Defined Terms

The following terms are used in this document:

Certificate: An X.509 Certificate, as specified in [[RFC5280](#)].

Certificate Evaluator: A party other than a Relying Party that evaluates the trustworthiness of certificates issued by Certification Authorities.

Certification Authority (CA): An Issuer that issues certificates in accordance with a specified Certificate Policy.

Certificate Policy (CP): Specifies the criteria that a Certification Authority undertakes to meet in its issue of certificates. See [[RFC3647](#)].

Certification Practices Statement (CPS): Specifies the means by which the criteria of the Certificate Policy are met. In most cases, this will be the document against which the operations of the Certification Authority are audited. See [[RFC3647](#)].

Domain Name: The label assigned to a node in the Domain Name System.

Domain Name System (DNS): The Internet naming system specified in [[RFC1034](#)] and [[RFC1035](#)].

DNS Security (DNSSEC): Extensions to the DNS that provide authentication services as specified in [[RFC4033](#)], [[RFC4034](#)], [[RFC4035](#)], [[RFC5155](#)], and revisions.

Fully-Qualified Domain Name: A Domain Name that includes the labels of all superior nodes in the Domain Name System.

Issuer: An entity that issues certificates. See [[RFC5280](#)].

Property: The tag-value portion of a CAA Resource Record.

Property Tag: The tag portion of a CAA Resource Record.

Property Value: The value portion of a CAA Resource Record.

Resource Record (RR): A particular entry in the DNS including the owner name, class, type, time to live, and data, as defined in [[RFC1034](#)] and [[RFC2181](#)].

Resource Record Set (RRSet): A set of Resource Records of a particular owner name, class, and type. The time to live on all RRs

with an RRSSet is always the same, but the data may be different among RRs in the RRSSet.

Relevant Resource Record Set (Relevant RRSSet): A set of CAA Resource Records resulting from applying the algorithm in [Section 4](#) to a specific Domain Name or Wildcard Domain Name.

Relying Party: A party that makes use of an application whose operation depends on use of a certificate for making a security decision. See [[RFC5280](#)].

Wildcard Domain Name: A Domain Name consisting of a single asterisk character followed by a single full stop character ("*.") followed by a Fully-Qualified Domain Name.

3. Relevant Resource Record Set

Before issuing a certificate, a compliant CA MUST check for publication of a Relevant RRSSet. If such an RRSSet exists, a CA MUST NOT issue a certificate unless the CA determines that either (1) the certificate request is consistent with the applicable CAA Resource Record set or (2) an exception specified in the relevant Certificate Policy or Certification Practices Statement applies. If the Relevant RRSSet for a Domain Name or Wildcard Domain Name contains no Property Tags that restrict issuance (for instance, if it contains only `iodef` Property Tags, or only Property Tags unrecognized by the CA), CAA does not restrict issuance.

A certificate request MAY specify more than one Domain Name and MAY specify Wildcard Domain Names. Issuers MUST verify authorization for all the Domain Names and Wildcard Domain Names specified in the request.

The search for a CAA RRSSet climbs the DNS name tree from the specified label up to but not including the DNS root '.' until a CAA RRSSet is found.

Given a request for a specific Domain Name X, or a request for a Wildcard Domain Name *.X, the Relevant Resource Record Set RelevantCAASet(X) is determined as follows:

Let CAA(X) be the RRSSet returned by performing a CAA record query for the Domain Name X, according to the lookup algorithm specified in [RFC 1034 section 4.3.2](#) (in particular chasing aliases). Let Parent(X) be the Domain Name produced by removing the leftmost label of X.


```

RelevantCAASet(domain):
  for domain is not ".":
    if CAA(domain) is not Empty:
      return CAA(domain)
    domain = Parent(domain)
  return Empty

```

For example, processing CAA for the Domain Name "X.Y.Z" where there are no CAA records at any level in the tree RelevantCAASet would have the following steps:

```

CAA("X.Y.Z.") = Empty; domain = Parent("X.Y.Z.") = "Y.Z."
CAA("Y.Z.")   = Empty; domain = Parent("Y.Z.")   = "Z."
CAA("Z.")     = Empty; domain = Parent("Z.")     = "."
return Empty

```

Processing CAA for the Domain Name "A.B.C" where there is a CAA record "issue example.com" at "B.C" would terminate early upon finding the CAA record:

```

CAA("A.B.C.") = Empty; domain = Parent("A.B.C.") = "B.C."
CAA("B.C.")   = "issue example.com"
return "issue example.com"

```

4. Mechanism

4.1. Syntax

A CAA Resource Record contains a single Property consisting of a tag-value pair. A Domain Name MAY have multiple CAA RRs associated with it and a given Property Tag MAY be specified more than once across those RRs.

The RDATA section for a CAA Resource Record contains one Property. A Property consists of the following:

```

+0-1-2-3-4-5-6-7-|0-1-2-3-4-5-6-7-|
| Flags           | Tag Length = n |
+-----+-----+...+-----+
| Tag char 0      | Tag char 1      |...| Tag char n-1 |
+-----+-----+...+-----+
+-----+-----+.....+-----+
| Value byte 0    | Value byte 1    |.....| Value byte m-1 |
+-----+-----+.....+-----+

```

Where n is the length specified in the Tag length field and m is the remaining octets in the Value field. They are related by $(m = d - n - 2)$ where d is the length of the RDATA section.

The fields are defined as follows:

Flags: One octet containing the following field:

Bit 0, Issuer Critical Flag: If the value is set to '1', the Property is critical. A Certification Authority MUST NOT issue certificates for any Domain Name where the Relevant RRSset for that Domain Name contains a CAA critical Property for an unknown or unsupported Property Tag.

Note that according to the conventions set out in [\[RFC1035\]](#), bit 0 is the Most Significant Bit and bit 7 is the Least Significant Bit. Thus, the Flags value 1 means that bit 7 is set while a value of 128 means that bit 0 is set according to this convention.

All other bit positions are reserved for future use.

To ensure compatibility with future extensions to CAA, DNS records compliant with this version of the CAA specification MUST clear (set to "0") all reserved flags bits. Applications that interpret CAA records MUST ignore the value of all reserved flag bits.

Tag Length: A single octet containing an unsigned integer specifying the tag length in octets. The tag length MUST be at least 1 and SHOULD be no more than 15.

Tag: The Property identifier, a sequence of US-ASCII characters.

Tags MAY contain US-ASCII characters 'a' through 'z', 'A' through 'Z', and the numbers 0 through 9. Tags SHOULD NOT contain any other characters. Matching of tags is case insensitive.

Tags submitted for registration by IANA MUST NOT contain any characters other than the (lowercase) US-ASCII characters 'a' through 'z' and the numbers 0 through 9.

Value: A sequence of octets representing the Property Value. Property Values are encoded as binary values and MAY employ sub-formats.

The length of the value field is specified implicitly as the remaining length of the enclosing RDATA section.

4.1.1. Canonical Presentation Format

The canonical presentation format of the CAA record is:

CAA <flags> <tag> <value>

Where:

Flags: Is an unsigned integer between 0 and 255.

Tag: Is a non-zero sequence of US-ASCII letters and numbers in lower case.

Value: The value field, expressed as a contiguous set of characters without interior spaces, or as a quoted string. See the the <character-string> format specified in [\[RFC1035\], Section 5.1](#), but note that the value field contains no length byte and is not limited to 255 characters.

4.2. CAA issue Property

If the issue Property Tag is present in the Relevant RRSSet for a Domain Name, it is a request that Issuers

1. Perform CAA issue restriction processing for the Domain Name, and
2. Grant authorization to issue certificates containing that Domain Name to the holder of the issuer-domain-name or a party acting under the explicit authority of the holder of the issuer-domain-name.

The CAA issue Property Value has the following sub-syntax (specified in ABNF as per [\[RFC5234\]](#)).

```
issue-value = *WSP [issuer-domain-name *WSP] [";" *WSP [parameters *WSP]]
```

```
issuer-domain-name = label *("." label)
```

```
label = (ALPHA / DIGIT) *( *("-") (ALPHA / DIGIT))
```

```
parameters = (parameter *WSP ";" *WSP parameters) / parameter
```

```
parameter = tag *WSP "=" *WSP value
```

```
tag = (ALPHA / DIGIT) *( *("-") (ALPHA / DIGIT))
```

```
value = *(%x21-3A / %x3C-7E)
```

For consistency with other aspects of DNS administration, Domain Name values are specified in letter-digit-hyphen Label (LDH-Label) form.

The following CAA record set requests that no certificates be issued for the Domain Name 'certs.example.com' by any Issuer other than ca1.example.net or ca2.example.org.

```
certs.example.com      CAA 0 issue "ca1.example.net"
certs.example.com      CAA 0 issue "ca2.example.org"
```


Because the presence of an issue Property Tag in the Relevant RRSset for a Domain Name restricts issuance, Domain Name owners can use an issue Property Tag with no issuer-domain-name to request no issuance.

For example, the following RRSset requests that no certificates be issued for the Domain Name 'nocerts.example.com' by any Issuer.

```
nocerts.example.com      CAA 0 issue ";"
```

An issue Property Tag where the issue-value does not match the ABNF grammar **MUST** be treated the same as one specifying an empty issuer-domain-name. For example, the following malformed CAA RRSset forbids issuance:

```
malformed.example.com    CAA 0 issue "%%%%%"
```

CAA authorizations are additive; thus, the result of specifying both an empty issuer-domain-name and a non-empty issuer-domain-name is the same as specifying just the non-empty issuer-domain-name.

An Issuer **MAY** choose to specify parameters that further constrain the issue of certificates by that Issuer, for example, specifying that certificates are to be subject to specific validation policies, billed to certain accounts, or issued under specific trust anchors.

For example, if cal.example.net has requested its customer accountable.example.com to specify their account number "230123" in each of the customer's CAA records using the (CA-defined) "account" parameter, it would look like this:

```
accountable.example.com  CAA 0 issue "cal.example.net; account=230123"
```

The semantics of parameters to the issue Property Tag are determined by the Issuer alone.

4.3. CAA issuewild Property

The issuewild Property Tag has the same syntax and semantics as the issue Property Tag except that it only grants authorization to issue certificates that specify a Wildcard Domain Name and issuewild properties take precedence over issue properties when specified. Specifically:

issuewild properties **MUST** be ignored when processing a request for a Domain Name (that is, not a Wildcard Domain Name).

If at least one `issuewild` Property is specified in the Relevant RRSset for a Wildcard Domain Name, all issue properties MUST be ignored when processing a request for that Wildcard Domain Name.

For example, the following RRSset requests that `_only_ ca1.example.net` issue certificates for "wild.example.com" or "sub.wild.example.com", and that `_only_ ca2.example.org` issue certificates for "*.wild.example.com" or "*.sub.wild.example.com").

```
wild.example.com      CAA 0 issue "ca1.example.net"
wild.example.com      CAA 0 issuewild "ca2.example.org"
```

The following RRSset requests that `_only_ ca1.example.net` issue certificates for "wild2.example.com", "*.wild2.example.com" or "*.sub.wild2.example.com".

```
wild2.example.com      CAA 0 issue "ca1.example.net"
```

The following RRSset requests that `_only_ ca2.example.org` issue certificates for "*.wild3.example.com" or "*.sub.wild3.example.com". It does not permit any Issuer to issue for "wild3.example.com" or "sub.wild3.example.com".

```
wild3.example.com      CAA 0 issuewild "ca2.example.org"
wild3.example.com      CAA 0 issue ";"
```

The following RRSset requests that `_only_ ca2.example.org` issue certificates for "*.wild3.example.com" or "*.sub.wild3.example.com". It permits any Issuer to issue for "wild3.example.com" or "sub.wild3.example.com".

```
wild3.example.com      CAA 0 issuewild "ca2.example.org"
```

4.4. CAA iodef Property

The `iodef` Property specifies a means of reporting certificate issue requests or cases of certificate issue for domains for which the Property appears in the Relevant RRSset, when those requests or issuances violate the security policy of the Issuer or the Domain Name holder.

The Incident Object Description Exchange Format (IODEF) [[RFC7970](#)] is used to present the incident report in machine-readable form.

The `iodef` Property Tag takes a URL as its Property Value. The URL scheme type determines the method used for reporting:

mailto: The IODEF incident report is reported as a MIME email attachment to an SMTP email that is submitted to the mail address specified. The mail message sent SHOULD contain a brief text message to alert the recipient to the nature of the attachment.

http or https: The IODEF report is submitted as a Web service request to the HTTP address specified using the protocol specified in [\[RFC6546\]](#).

The following RRSset specifies that reports may be made by means of email with the IODEF data as an attachment, a Web service [\[RFC6546\]](#), or both:

report.example.com	CAA 0 issue "ca1.example.net"
report.example.com	CAA 0 iodef "mailto:security@example.com"
report.example.com	CAA 0 iodef "http://iodef.example.com/"

4.5. Critical Flag

The critical flag is intended to permit future versions of CAA to introduce new semantics that MUST be understood for correct processing of the record, preventing conforming CAs that do not recognize the new semantics from issuing certificates for the indicated Domain Names.

In the following example, the Property with a Property Tag of 'tbs' is flagged as critical. Neither the ca1.example.net CA nor any other Issuer is authorized to issue for "new.example.com" (or any other domains for which this is the Relevant RRSset) unless the Issuer has implemented the processing rules for the 'tbs' Property Tag.

new.example.com	CAA 0 issue "ca1.example.net"
new.example.com	CAA 128 tbs "Unknown"

5. Security Considerations

CAA records assert a security policy that the holder of a Domain Name wishes to be observed by Issuers. The effectiveness of CAA records as an access control mechanism is thus dependent on observance of CAA constraints by Issuers.

The objective of the CAA record properties described in this document is to reduce the risk of certificate mis-issue rather than avoid reliance on a certificate that has been mis-issued. DANE [\[RFC6698\]](#) describes a mechanism for avoiding reliance on mis-issued certificates.

5.1. Use of DNS Security

Use of DNSSEC to authenticate CAA RRs is strongly RECOMMENDED but not required. An Issuer MUST NOT issue certificates if doing so would conflict with the Relevant RRSets, irrespective of whether the corresponding DNS records are signed.

DNSSEC provides a proof of non-existence for both DNS Domain Names and RRSets within Domain Names. DNSSEC verification thus enables an Issuer to determine if the answer to a CAA record query is empty because the RRSets is empty or if it is non-empty but the response has been suppressed.

Use of DNSSEC allows an Issuer to acquire and archive a proof that they were authorized to issue certificates for the Domain Name. Verification of such archives MAY be an audit requirement to verify CAA record processing compliance. Publication of such archives MAY be a transparency requirement to verify CAA record processing compliance.

5.2. Non-Compliance by Certification Authority

CAA records offer CAs a cost-effective means of mitigating the risk of certificate mis-issue: the cost of implementing CAA checks is very small and the potential costs of a mis-issue event include the removal of an embedded trust anchor.

5.3. Mis-Issue by Authorized Certification Authority

Use of CAA records does not prevent mis-issue by an authorized Certification Authority, i.e., a CA that is authorized to issue certificates for the Domain Name in question by CAA records.

Domain Name holders SHOULD verify that the CAs they authorize to issue certificates for their Domain Names employ appropriate controls to ensure that certificates are issued only to authorized parties within their organization.

Such controls are most appropriately determined by the Domain Name holder and the authorized CA(s) directly and are thus out of scope of this document.

5.4. Suppression or Spoofing of CAA Records

Suppression of the CAA record or insertion of a bogus CAA record could enable an attacker to obtain a certificate from an Issuer that was not authorized to issue for that Domain Name.

Where possible, Issuers SHOULD perform DNSSEC validation to detect missing or modified CAA record sets.

In cases where DNSSEC is not deployed for a corresponding Domain Name, an Issuer SHOULD attempt to mitigate this risk by employing appropriate DNS security controls. For example, all portions of the DNS lookup process SHOULD be performed against the authoritative name server. Data cached by third parties MUST NOT be relied on but MAY be used to support additional anti-spoofing or anti-suppression controls.

5.5. Denial of Service

Introduction of a malformed or malicious CAA RR could in theory enable a Denial-of-Service (DoS) attack.

This specific threat is not considered to add significantly to the risk of running an insecure DNS service.

An attacker could, in principle, perform a DoS attack against an Issuer by requesting a certificate with a maliciously long DNS name. In practice, the DNS protocol imposes a maximum name length and CAA processing does not exacerbate the existing need to mitigate DoS attacks to any meaningful degree.

5.6. Abuse of the Critical Flag

A Certification Authority could make use of the critical flag to trick customers into publishing records that prevent competing Certification Authorities from issuing certificates even though the customer intends to authorize multiple providers.

In practice, such an attack would be of minimal effect since any competent competitor that found itself unable to issue certificates due to lack of support for a Property marked critical SHOULD investigate the cause and report the reason to the customer. The customer will thus discover that they had been deceived.

6. Deployment Considerations

A CA implementing CAA may find that they receive errors looking up CAA records. The following are some common causes of such errors, so that CAs may provide guidance to their subscribers on fixing the underlying problems.

6.1. Blocked Queries or Responses

Some middleboxes, in particular anti-DDoS appliances, may be configured to drop DNS packets of unknown types, or may start dropping such packets when they consider themselves under attack. This generally manifests as a timed-out DNS query, or a SERVFAIL at a local recursive resolver.

6.2. Rejected Queries and Malformed Responses

Some authoritative nameservers respond with REJECTED or NOTIMP when queried for a Resource Record type they do not recognize. At least one authoritative resolver produces a malformed response (with the QR bit set to 0) when queried for unknown Resource Record types. Per [RFC 1034](#), the correct response for unknown Resource Record types is NOERROR.

6.3. Delegation to Private Nameservers

Some Domain Name administrators make the contents of a subdomain unresolvable on the public Internet by delegating that subdomain to a nameserver whose IP address is private. A CA processing CAA records for such subdomains will receive SERVFAIL from its recursive resolver. The CA MAY interpret that as preventing issuance. Domain Name administrators wishing to issue certificates for private Domain Names SHOULD use split-horizon DNS with a publicly available nameserver, so that CAs can receive a valid, empty CAA response for those Domain Names.

6.4. Bogus DNSSEC Responses

Queries for CAA Resource Records are different from most DNS RR types, because a signed, empty response to a query for CAA RRs is meaningfully different from a bogus response. A signed, empty response indicates that there is definitely no CAA policy set at a given label. A bogus response may mean either a misconfigured zone, or an attacker tampering with records. DNSSEC implementations may have bugs with signatures on empty responses that go unnoticed, because for more common Resource Record types like A and AAAA, the difference to an end user between empty and bogus is irrelevant; they both mean a site is unavailable.

In particular, at least two authoritative resolvers that implement live signing had bugs when returning empty Resource Record sets for DNSSEC-signed zones, in combination with mixed-case queries. Mixed-case queries, also known as DNS 0x20, are used by some recursive resolvers to increase resilience against DNS poisoning attacks. DNSSEC-signing authoritative resolvers are expected to copy the same

capitalization from the query into their ANSWER section, but sign the response as if they had use all lowercase. In particular, PowerDNS versions prior to 4.0.4 had this bug.

7. Differences versus [RFC6844](#)

This document obsoletes [RFC6844](#). The most important change is to the Certification Authority Processing section. [RFC6844](#) specified an algorithm that performed DNS tree-climbing not only on the Domain Name being processed, but also on all CNAMEs and DNAMEs encountered along the way. This made the processing algorithm very inefficient when used on Domain Names that utilize many CNAMEs, and would have made it difficult for hosting providers to set CAA policies on their own Domain Names without setting potentially unwanted CAA policies on their customers' Domain Names. This document specifies a simplified processing algorithm that only performs tree climbing on the Domain Name being processed, and leaves processing of CNAMEs and DNAMEs up to the CA's recursive resolver.

This document also includes a "Deployment Considerations" section detailing experience gained with practical deployment of CAA enforcement among CAs in the WebPKI.

This document clarifies the ABNF grammar for issue and issuewild tags and resolves some inconsistencies with the document text. In particular, it specifies that parameters are separated with semicolons. It also allows hyphens in Property Tags.

This document also clarifies processing of a CAA RRset that is not empty, but contains no issue or issuewild tags.

This document removes the section titled "The CAA RR Type," merging it with "Mechanism" because the definitions were mainly duplicates. It moves the "Use of DNS Security" section into Security Considerations. It renames "Certification Authority Processing" to "Relevant Resource Record Set," and emphasizes the use of that term to more clearly define which domains are affected by a given RRset.

8. IANA Considerations

IANA is requested to add [[[RFC Editor: Please replace with this RFC]]] as a reference for the Certification Authority Restriction Flags and Certification Authority Restriction Properties registries.

9. Acknowledgements

The authors would like to thank the following people who contributed to the design and documentation of this work item: Corey Bonnell, Chris Evans, Stephen Farrell, Jeff Hodges, Paul Hoffman, Tim Hollebeek, Stephen Kent, Adam Langley, Ben Laurie, James Manger, Chris Palmer, Scott Schmit, Sean Turner, and Ben Wilson.

10. References

10.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", [RFC 2181](#), DOI 10.17487/RFC2181, July 1997, <<https://www.rfc-editor.org/info/rfc2181>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", [RFC 4034](#), DOI 10.17487/RFC4034, March 2005, <<https://www.rfc-editor.org/info/rfc4034>>.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", [RFC 4035](#), DOI 10.17487/RFC4035, March 2005, <<https://www.rfc-editor.org/info/rfc4035>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", [RFC 5155](#), DOI 10.17487/RFC5155, March 2008, <<https://www.rfc-editor.org/info/rfc5155>>.
- [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>>.

- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", [RFC 6546](#), DOI 10.17487/RFC6546, April 2012, <<https://www.rfc-editor.org/info/rfc6546>>.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", [RFC 6698](#), DOI 10.17487/RFC6698, August 2012, <<https://www.rfc-editor.org/info/rfc6698>>.
- [RFC7970] Danyliw, R., "The Incident Object Description Exchange Format Version 2", [RFC 7970](#), DOI 10.17487/RFC7970, November 2016, <<https://www.rfc-editor.org/info/rfc7970>>.
- [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>>.

[10.2](#). Informative References

- [RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", [RFC 3647](#), DOI 10.17487/RFC3647, November 2003, <<https://www.rfc-editor.org/info/rfc3647>>.

Authors' Addresses

Phillip Hallam-Baker

Email: phill@hallambaker.com

Rob Stradling
Sectigo Ltd.

Email: rob@sectigo.com

Jacob Hoffman-Andrews
Let's Encrypt

Email: jsha@letsencrypt.org