Network Working Group
Internet-Draft
Obsoletes: RFC 6844 (if approved)
Intended status: Standards Track
Expires: April 13, 2019

P. Hallam-Baker
R. Stradling
Comodo Group, Inc
J. Hoffman-Andrews
Let's Encrypt

October 10, 2018

DNS Certification Authority Authorization (CAA) Resource Record
draft-ietf-lamps-rfc6844bis-01

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. 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.

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 April 13, 2019.

Copyright Notice

Table of Contents

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. 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 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. Before issuing a certificate, a PKIX CA is required to validate the request according to the policies set out in its Certificate Policy. In the case of a public CA that validates certificate requests as a third party, the certificate will typically be issued under a public trust anchor certificate embedded in one or more relevant Relying Applications.

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 Certificate 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. Since a certificate is typically valid for at least a year, 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 Applications 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 [RFC2119].

2.2. Defined Terms

The following terms are used in this document:

Authorization Entry: An authorization assertion that grants or denies a specific set of permissions to a specific group of entities.

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: A DNS Domain Name.

Domain Name: A DNS Domain Name as specified in [STD13].

Domain Name System (DNS): The Internet naming system specified in [STD13].

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

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.

Public Key Infrastructure X.509 (PKIX): Standards and specifications issued by the IETF that apply the [X.509] certificate standards specified by the ITU to Internet applications as specified in [RFC5280] and related documents.

Resource Record (RR): A particular entry in the DNS including the owner name, class, type, time to live, and data, as defined in [STD13] and [RFC2181].

Resource Record Set (RRSet): A set of Resource Records or a particular owner name, class, and type. The time to live on all RRs with an RRSet is always the same, but the data may be different among RRs in the RRSet.

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].

Relying Application: An application whose operation depends on use of a certificate for making a security decision.

3. The CAA RR Type

A CAA RR consists of a flags byte and a tag-value pair referred to as a property. Multiple properties MAY be associated with the same domain name by publishing multiple CAA RRs at that domain name. The following flag is defined:

Issuer Critical: If set to '1', indicates that the corresponding property tag MUST be understood if the semantics of the CAA record are to be correctly interpreted by an issuer.

Issuers MUST NOT issue certificates for a domain if the relevant CAA Resource Record set contains unknown property tags that have the Critical bit set.

The following property tags are defined:

issue <Issuer Domain Name> [; <name>=<value>]* : The issue property entry authorizes the holder of the domain name <Issuer Domain Name> or a party acting under the explicit authority of the holder of that domain name to issue certificates for the domain in which the property is published.

issuewild <Issuer Domain Name> [; <name>=<value>]* : The issuewild property entry authorizes the holder of the domain name <Issuer Domain Name> or a party acting under the explicit authority of the holder of that domain name to issue wildcard certificates for the domain in which the property is published.

iodef <URL> : Specifies a URL to which an issuer MAY report certificate issue requests that are inconsistent with the issuer's Certification Practices or Certificate Policy, or that a Certificate

Evaluator may use to report observation of a possible policy violation. The Incident Object Description Exchange Format (IODEF) format is used [RFC5070].

The following example is a DNS zone file (see [RFC1035]) that informs CAs that certificates are not to be issued except by the holder of the domain name 'ca.example.net' or an authorized agent thereof. This policy applies to all subordinate domains under example.com.

```
$ORIGIN example.com
.       CAA 0 issue "ca.example.net"
```

If the domain name holder specifies one or more iodef properties, a certificate issuer MAY report invalid certificate requests to that address. In the following example, the domain name holder specifies that reports may be made by means of email with the IODEF data as an attachment, a Web service [RFC6546], or both:

```
$ORIGIN example.com
.       CAA 0 issue "ca.example.net"
.       CAA 0 iodef "mailto:security@example.com"
.       CAA 0 iodef "http://iodef.example.com/"
```

A certificate issuer MAY specify additional parameters that allow customers to specify additional parameters governing certificate issuance. This might be the Certificate Policy under which the certificate is to be issued, the authentication process to be used might be specified, or an account number specified by the CA to enable these parameters to be retrieved.

For example, the CA 'ca.example.net' has requested its customer 'example.com' to specify the CA's account number '230123' in each of the customer's CAA records.

```
$ORIGIN example.com
.       CAA 0 issue "ca.example.net; account=230123"
```

The syntax of additional parameters is a sequence of name-value pairs as defined in Section 5.2. The semantics of such parameters is left to site policy and is outside the scope of this document.

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 domains.

In the following example, the property 'tbs' is flagged as critical. Neither the example.net CA nor any other issuer is authorized to issue under either policy unless the processing rules for the 'tbs' property tag are understood.

```
$ORIGIN example.com
.       CAA 0 issue "ca.example.net; policy=ev"
.       CAA 128 tbs "Unknown"
```

Note that the above restrictions only apply at certificate issue. Since the validity of an end entity certificate is typically a year or more, it is quite possible that the CAA records published at a domain will change between the time a certificate was issued and validation by a relying party.

4. Certification Authority Processing

Before issuing a certificate, a compliant CA MUST check for publication of a relevant CAA Resource Record set. If such a record set 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.

A certificate request MAY specify more than one domain name and MAY specify wildcard domains. Issuers MUST verify authorization for all the domains and wildcard domains specified in the request.

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

Given a request for a specific domain name X, or a request for a wildcard domain name *.X, the relevant record set RelevantCAASet(X) is determined as follows:

Let CAA(X) be the record set 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"
```

4.1. 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 CAA Resource Record set, irrespective of whether the corresponding DNS records are signed.

DNSSEC provides a proof of non-existence for both DNS domains and RR set within domains. DNSSEC verification thus enables an issuer to determine if the answer to a CAA record query is empty because the RR set 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. 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.

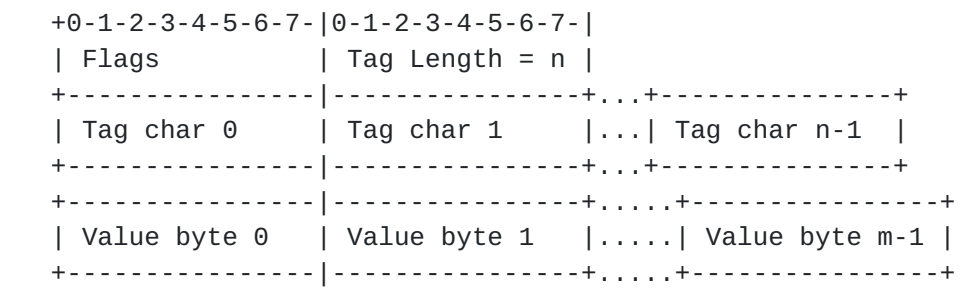
5. Mechanism

5.1. Syntax

A CAA RR contains a single property entry consisting of a tag-value pair. Each tag represents a property of the CAA record. The value of a CAA property is that specified in the corresponding value field.

A domain name MAY have multiple CAA RRs associated with it and a given property MAY be specified more than once.

The CAA data field contains one property entry. A property entry consists of the following data fields:



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

The data fields are defined as follows:

Flags: One octet containing the following fields:

Bit 0, Issuer Critical Flag: If the value is set to '1', the critical flag is asserted and the property MUST be understood if the CAA record is to be correctly processed by a certificate issuer.

A Certification Authority MUST NOT issue certificates for any Domain that contains a CAA critical property for an unknown or unsupported property tag that for which the issuer critical flag is set.

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.

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

Tag values 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 Resource Record data field.

5.1.1. 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: Is the <character-string> encoding of the value field as specified in [RFC1035], Section 5.1.

5.2. CAA issue Property

The issue property tag is used to request that certificate issuers perform CAA issue restriction processing for the domain and to grant authorization to specific certificate issuers.

The CAA issue property value has the following sub-syntax (specified in ABNF as per [RFC5234]).

```
issuevalue = *WSP [domain *WSP] [";" *WSP [parameters *WSP]]

domain = 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.

A CAA record with an issue parameter tag that does not specify a domain name is a request that certificate issuers perform CAA issue restriction processing for the corresponding domain without granting authorization to any certificate issuer.

This form of issue restriction would be appropriate to specify that no certificates are to be issued for the domain in question.

For example, the following CAA resource record set requests that no certificates be issued for the domain 'nocerts.example.com' by any certificate issuer.

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

A CAA record with an issue parameter tag that specifies a domain name is a request that certificate issuers perform CAA issue restriction processing for the corresponding domain and grants authorization to the certificate issuer specified by the domain name.

For example, the following CAA record set requests that no certificates be issued for the domain 'certs.example.com' by any certificate issuer other than the example.net certificate issuer.

```
certs.example.com CAA 0 issue "example.net"
```

CAA authorizations are additive; thus, the result of specifying both the empty issuer and a specified issuer is the same as specifying just the specified issuer alone.

An issue property tag where the issuevalue does not match the ABNF grammar MUST be treated the same as one specifying the empty issuer.

For example, the following malformed CAA resource record set forbids issuance:

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

A non-empty CAA record set that contains no issue property tags is authorization to any certificate issuer to issue for the corresponding domain, provided that it is a non-wildcard domain, and no records in the CAA record set otherwise prohibit issuance.

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

The semantics of issuer-parameters are determined by the issuer alone.

5.3. CAA issuewild Property

The issuewild property has the same syntax and semantics as the issue property except that issuewild properties only grant authorization to issue certificates that specify a wildcard domain and issuewild properties take precedence over issue properties when specified. Specifically:

issuewild properties MUST be ignored when processing a request for a domain that is not a wildcard domain.

If at least one issuewild property is specified in the relevant CAA record set, all issue properties MUST be ignored when processing a request for a domain that is a wildcard domain.

A non-empty CAA record set that contains no issue or issuewild property tags is authorization to any certificate issuer to issue for the corresponding wildcard domain, provided that no records in the CAA record set otherwise prohibit issuance.

5.4. CAA iodef Property

The iodef property specifies a means of reporting certificate issue requests or cases of certificate issue for the corresponding domain that violate the security policy of the issuer or the domain name holder.

The Incident Object Description Exchange Format (IODEF) [RFC5070] is used to present the incident report in machine-readable form.

The iodef property takes a URL as its parameter. 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].

6. Security Considerations

CAA records assert a security policy that the holder of a domain name wishes to be observed by certificate 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.

6.1. 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.

6.2. 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 in question by CAA records.

Domain name holders SHOULD verify that the CAs they authorize to issue certificates for their domains 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.

6.3. 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 in a corresponding domain, 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.

6.4. 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.

6.5. 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.

7. Deployment Considerations

7.1. 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.

For deployability of CAA and future DNS record types, middleboxes SHOULD block DNS packets by volume and size rather than by query type.

7.2. 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.

7.3. Delegation to Private Nameservers

Some domain 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 administrators wishing to issue certificates for private domains SHOULD use split-horizon DNS with a publicly available nameserver, so that CAs can receive a valid, empty CAA response for those domains.

7.4. 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.

8. 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 domains that utilize many CNAMEs, and would have made it difficult for hosting providers to set CAA policies on their own domains without setting potentially unwanted CAA policies on their customers' domains. This document specifies a simplified processing algorithm that only performs tree climbing on the domain 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 amount 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 hyphens. It also allows hyphens in property names.

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

9. IANA Considerations

This document has no IANA actions.

9.1. Certification Authority Restriction Flags

IANA has created the "Certification Authority Restriction Flags" registry with the following initial values:

Flag	Meaning	Reference
0	Issuer Critical Flag	[RFC6844]
1-7	Reserved>	[RFC6844]

Assignment of new flags follows the RFC Required policy set out in [RFC5226], Section 4.1.

10. Acknowledgements

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

11. Normative References

[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>.

[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>.

[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>.

[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>.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <https://www.rfc-editor.org/info/rfc5070>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[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>.

[RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 6844, DOI 10.17487/RFC6844, January 2013, <https://www.rfc-editor.org/info/rfc6844>.

Authors' Addresses

Phillip Hallam-Baker
Comodo Group, Inc

Email: philliph@comodo.com

Rob Stradling
Comodo Group, Inc

Email: rob.stradling@comodo.com

Jacob Hoffman-Andrews
Let's Encrypt

Email: jsha@letsencrypt.org