Limited Additional Mechanisms for PKIX and SMIME
Internet-Draft
Intended status: Standards Track
Expires: 22 September 2025

H. Birge-Lee
G. Cimaszewski
C. E. Kr辰henb端hl
L. Wang
Princeton University
A. Gable
ISRG
P. Mittal
Princeton University

21 March 2025

CAA Security Tag for Cryptographically-Constrained Domain Validation
draft-birgelee-lamps-caa-security-02

Abstract

Cryptographic domain validation procedures leverage authenticated communication channels to ensure resilience against attacks by both on-path and off-path network adversaries which may be located between the Certification Authority (CA) and the network resources related to the domain contained in the certificate. Domain owners can leverage "security" Property Tags specified in CAA records (defined in [RFC8659]) with the critical flag set, to ensure that CAs perform cryptographically-constrained domain validation during their issuance procedure, hence defending against global man-in-the-middle adversaries. This document defines the syntax of the CAA security Property as well as acceptable means for cryptographically-constrained domain validation procedures.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://birgelee.github.io/draft-caa-security-tag/draft-birgelee-lamps-caa-security.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-birgelee-lamps-caa-security/.

Discussion of this document takes place on the Limited Additional Mechanisms for PKIX and SMIME Working Group mailing list (mailto:spasm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/spasm/. Subscribe at https://www.ietf.org/mailman/listinfo/spasm/.

Source for this draft and an issue tracker can be found at https://github.com/birgelee/draft-caa-security-tag.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A CAA security Property Tag is compliant with [RFC8659] and puts restrictions on the circumstances under which a CA is allowed to sign a certificate for a given domain. A security Property Tag on a domain implies that validation for this domain must be done in a manner that defends against network adversaries even if an adversary is capable of intercepting and/or modifying communication between the CA and the network resources related to the domain being validated. Issuance of a certificate to a domain with a security Property Tag MUST follow one of the specified Cryptographically-constrained Domain Validation (CDV) methods outlined in this document or future extensions. CDV methods MUST rely on protocols (like DNSSEC or DoH/DoT) that offer security properties even in the presence of man-in-the-middle adversaries that can intercept any communication occurring over the public Internet.

Not all CDV methods are themselves compliant with the CA/Browser Forum's Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates. Hence, any CDV method that does not meet the CA/Browser Forum Baseline Requirements for TLS server certificate issuance must be used in conjunction with such a compliant domain validation method.

2. Conventions and Definitions

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.

2.1. Cryptographic Domain Validation

The goal of cryptographically-constrained domain validation is to ensure that domain validation is based on communication channels that are authenticated through cryptographic operations.

2.1.1. Threat Model

Cryptographic domain validation defends against a network adversary that can block, intercept, modify, and forge arbitrary network packets sent between a CA and the domain's network resources, but cannot forge cryptographic objects, such as signatures and encrypted data. Cryptographic domain validation is based on DNS and thus assumes that a domain owner can securely mange their DNS records, i.e., communication between the domain owner and their DNS infrastructure is protected against network adversaries. Similarly, communication between the entity generating the private key and requesting the certificate, e.g., the domain owner, and the webserver(s) where the private key and certificate are installed, is assumed to be protected against network adversaries. Furthermore, it assumes that all CAs are benign and correctly follow all necessary validation procedures as described in the relevant standardization documents.

Cryptographic domain validation can be used on domains that are contained in both domain validation certificates (where only the domain name in a certificate is validated) and extended or organization validated certificates (where information like organization identity as well as domain name is validated). Cryptographic domain validation only hardens the security of the validation of domain names, not broader identities (e.g., organization names). The use of cryptographically-constrained domain validation in an OV or EV certificate only improves the validation of the domain name(s) contained in the certificate (in the common name or subject-alternate names fields) and does not impact the validation of other forms of identity contained in the certificate. Use of cryptographically-constrained domain validation in a DV certificate does not imply validation of any identity beyond the domain name(s) in the certificate.

The defense involves the domain owner specifying a policy indicating their desire for cryptographically-constrained domain validation via DNS CAA records and securely communicating these records to all CAs. Hence, a core aspect of cryptographically-constrained domain validation is 1) ensuring secure policy lookups, and 2) preventing downgrade attacks that convince a CA to issue a certificate using non-cryptographically-constrained domain validation.

2.1.2. Secure Policy Lookup

The authenticity of the retrieved security CAA record SHOULD be protected to prevent an active network adversary from modifying its content. Authenticity can either be ensured by signing the security CAA record or by retrieving the security CAA record via an authenticated channel. Any security CAA record not protected by such a signature or authenticated channel may not benefit from the security properties outlined in this document.

2.1.2.1. Signed Record

A security CAA record SHOULD be protected with a valid DNSSEC signature chain going back to the ICANN DNSSEC root, to prove the authenticity of the record and its content.

2.1.2.2. Authenticated Channel

If it is not possible to have a DNSSEC signature chain back to the ICANN root, security CAA records SHOULD alternately be hosted on an authoritative DNS resolver that supports recursive-to-authoritative authenticated channels. Authenticated channels between recursive and authoritative nameservers could be deployed as described in [RFC9539] and leverage DoT, DoQ, or DoH as protocols providing an authenticated channel. Since secure policy lookup considers a more stringent threat model than the passive network adversary in [RFC9539], the deployment MUST also implement defenses against active adversaries highlighted in Appendix B of [RFC9539]. One option to implement these defenses is by creating DNSSEC-protected SVCB DNS records at the authoritative nameserver. These SVCB records provide a signaling mechanism for authoritative nameservers offering authenticated channel. Furthermore, the authenticity of the TLS certificate public key used to establish the channel MUST be protected, e.g., by specifying the public key hash as an SVCB parameter. This step is crucial to achieve our desired security properties, since it eliminates the circular dependency of requiring an authentic TLS certificate to secure the issuance of new TLS certificate.

2.1.3. Downgrade Prevention

To ensure that the CAA security Property is immediately and incrementally deployable without requiring all publicly-trusted CAs to understand the property, the CAA record containing the property MUST set the critical flag.

Serving security CAA records over authenticated DNS channels or using authenticated DNS records (i.e., DNSSEC) is critical to the effectiveness of the records because a security CAA record not

protected by authenticated DNS may be suppressed by an adversary that can manipulate DNS responses. This could potentially allow the adversary to downgrade validation to use a low-security method and undermine the security properties of the security Property Tag.

If DNSSEC is used to authenticate the CAA record, a CA MUST only accept the non-existence of a security CAA record if its non-existence is proven by NSEC record as described in [RFC7129].

If authenticated channels are used to authenticate the CAA record, CAs MUST also require recursive-to-authoritative DoT or DoH communication (and not permit standard unencrypted DNS connections) for DNS providers that host security CAA records. This prevents downgrade attacks where an adversary attempts to interfere with the establishment of a DoT or DoH encrypted channel and cause a fallback to unencrypted DNS over UDP or TCP.

3. CAA security Property

The CAA security Property Tag MUST be "security" and the flags field of a CAA record containing the security Property MUST have the critical bit set. In this document, we refer to a CAA record with these characteristics as a *security CAA record*.

3.1. Syntax

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

```
security-value = *WSP [attribute-list] *WSP

attribute-list = (attribute *WSP ";" *WSP attribute-list) / attribute
attribute = attribute-name *WSP "=" *WSP attribute-value

attribute-name = (ALPHA / DIGIT) *( *("-") (ALPHA / DIGIT))
attribute-value = *(value-char / WSP) value-char *(value-char / WSP)
value-char = %x21-3A / %x3C-7E
```

Hence, the security Property Value can either be empty or entirely whitespace, or contain a list of semicolon-separated attribute name-value pairs.

Similar to [RFC8659], attribute names are specified in letter-digit-hyphen Label (LDH-Label) form while attribute values can contain whitespace and any printable character except semicolon. Note that attribute values MUST contain at least one printable (non-whitespace) character.

All attributes specified in an attribute-list MUST be unique. An attribute-list MUST NOT have two attributes with the same name specified even if they contain different attribute values.

3.2. Well-known Attributes

The attribute-list MAY contain the following attributes.

The attribute values of the attributes specified in this document have the following sub-syntax (specified in ABNF as per [RFC5234]).

```
well-known-attribute-value = *WSP comma-sep-list *WSP

comma-sep-list = (list-item *WSP "," *WSP comma-sep-list) / list-item
list-item = 1*item-char
item-char = %x21-2B / %x2D-3A / %x3C-7E
```

1. *methods:* If specified, this attribute MUST have a non-empty comma-separated list of cryptographically-constrained domain validation methods that can be used to validate that particular domain. A CA MUST use one of the methods specified in the methods attribute value to perform cryptographically-constrained domain validation. If there is no method specified that the CA is capable of complying with, the CA MUST deny issuance.

2. *options:* If specified, this attribute MUST have a non-empty comma-separated list of options. A CA SHOULD try to honor any option specified in this list. If a CA does not understand an option or does not have that option implemented the, CA MAY proceed with issuance.

3. *options-critical:* If specified, this attribute MUST have a non-empty comma-separated list of options. To proceed with issuance, a CA MUST understand and implement all options specified in the options-critical attribute value.

The attribute-list MAY contain additional attributes and a CA MAY proceed with issuance even if it does not understand these additional attributes. Subsequent RFCs MAY standardize additional attributes.

3.2.1. Permissible Methods

The following attributes MAY be specified in the methods attribute value. Each method specifies particular aspects of certificate issuance that MUST be satisfied for a certificate to be issued using that method. While some methods entail the use of CA/Browser Forum-compliant domain control validation methods, others do not entail CA/Browser Forum-compliant domain control validation and must be used in

conjunction with a CA/Browser Forum-compliant domain control validation method to permit certificate issuance.

1. *secure-dns-record-change:* This method involves an applicant showing control of a DNSSEC-protected DNS record or a record that was retrieved via a DoH or DoT tunnel to the relevant authoritative nameservers used in the DNS resolution. This can be done via 1) the validation method "DNS Change" specified in the CA/Browser Forum's Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates (Section 3.2.2.4.7) or 2) the "dns-01" method of the ACME RFC [RFC8555]. For this method to be satisfied, the FQDN where the DNS change is demonstrated MUST be protected by DNSSEC or lookups to the relevant authoritative nameservers MUST be conducted over authenticated channels (e.g., DoH/DoT).

2. *http-validation-over-tls:* This method involves the completion of an HTTP domain validation challenge over an HTTPS session using TCP port 443 where the server authenticates with an existing publicly-trusted valid certificate covering the domain in question. The certificate cannot be self-signed or expired. This method MAY be directly satisfied while a CA is performing the "Agreed-Upon Change to Website v2" domain control validation method specified in the the CA/Browser Forum's Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates (Section 3.2.2.4.18). The ACME "http-01" challenge specified in [RFC8555] does not permit the use of HTTPS or port 443 when a CA is contacting the domain in question. A CA MAY still satisfy the *http-validation-over-tls* method even if it does not initiate connections to port 443 for HTTP challenges so long as either 1) the connection initiated to port 80 serves a redirect to the same domain name over HTTPS at port 443 and the connection to the domain at port 443 servers a valid, trusted certificate or 2) in addition to contacting the domain over port 80 the CA also contacts the domain over port 443 using HTTPS and the connection is established with a valid, trusted certificate and the same challenge value is observed. Operators of security-critical domains MAY choose not to permit this method since, unlike other cryptographically-constrained domain validation methods specified in this document, its security relies on the non-existence of malicious certificates for a domain at time of the creation of the security Property Tag in the domain's policy.

3. *known-account-specifier:* For a CA to issue a certificate using this method 1) there MUST exist a unique identifier for a CA subscriber account that is communicated with the CA out-of-band, over authenticated DNS lookups, or in another manner that is immune to man-in-the-middle adversaries, and 2) the CA may only

issue a certificate to an applicant that has authenticated itself to the CA as having access to that specified subscriber account. A CA does not have permission to issue under this method unless both of these criteria are met. Once these criteria have been met, the CA MUST additionally perform a validation method that is compliant with the Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates. One acceptable way of including this account identifier is with the CAA ACME account URI extension, defined in [RFC8657], in an authenticated DNS record.

4. *private-key-control:* This method involves an applicant showing control of a private key that corresponds to a public key placed in a DNS record associated with the domain being validated. The private key must be used to sign a message containing: a unique identifier for the CA, the domain name(s) in the certificate, a timestamp, and a hash of the public key in the certificate. This message may be hashed and then have the signature generated over the hash of this message. Obtaining such a signed message from a certificate applicant authorizes the CA specified in the message to sign a certificate for those domain names with the specified public key within 24h of the timestamp provided in the message. The CA MUST retrieve the public key or a hash of the public key corresponding to the private key used for signing the message via an authenticated DNS lookup using either authenticated channels to the relevant authoritative nameservers (e.g., DoH or DoT) or validation of a DNSSEC signature chain back to the ICANN root. After private key control is established, the CA MUST additionally perform a validation method that is compliant with the Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates.

In the event that *no methods attribute is specified in the attribute-list,* all methods specified in this document are acceptable as well as cryptographically-constrained domain validation methods defined in future RFCs. Future RFCs MAY specify additional methods for cryptographically-constrained domain validation so long as they satisfy the properties of cryptographically-constrained domain validation (i.e., robustness against global man-in-the-middle adversaries).

3.2.2. Permissible Options

The following options MAY be used in the options or options-critical attribute values.

1. *authenticated-policy-retrieval:* This option signifies to a CA that it MUST retrieve a domain's CAA security Property and any associated domain-owner identity (e.g., identifiers used for known-account-specifier and private-key-control) using authenticated DNS lookups or other authenticated channels. If a CA finds this option in the options-critical attribute and the CAA security Property was not retrieved using authenticated DNS lookups, the CA MUST NOT issue a certificate for that domain.

Additionally, a CA MAY choose to honor its own non-standardized options that do not need to be supported by other CAs or the IETF. These options MUST be prefixed with "ca-<ca_name>-" where ca_name is the name of the CA that initially developed the option.

3.3. Co-existence with other CAA Properties

The behavior of a CA when encountering a CAA RRset that contains multiple CAA Properties, including a security Property, depends on the CAA Property Tags.

3.3.1. CAA security Property

To minimize complexity and avoid the risk of unexpected behavior, a domain's entire cryptographically-constrained domain validation policy SHOULD be encoded into a single CAA security Property. If a CAA RRset contains multiple security Properties, a CA MUST block issuance if the certificate request does not comply with all of the security Tags. This ensures that if a new security Property Tag is specified, its security properties cannot be subverted by a stale security Property Tag.

3.3.2. CAA issue and issuewild Property

If a domain specifies both security Properties and a set of issue and issuewild Properties, the CA MUST adhere to all security Properties, as described above, and the CA MUST adhere to the set of issue and issuewild Properties as described in [RFC8659].

3.3.3. CAA iodef Property

The usage of the iodef Property is analogous to [RFC8659], i.e., it provides a CA the means of reporting certificate issue requests or cases of certificate issue for domains for which the Property appears in the Relevant RRset, when those requests or issuances violate the security policy of the Issuer or the FQDN holder. The iodef Property can be used to inform a domain owner about a blocked issuance due to an overly restrictive security Property.

4. Security Considerations

Many of the security considerations regarding security CAA records are inherited from those of CAA records more generally. Because security CAA records do not introduce any new methods for validating domain ownership, they do not increase the attack surface of fraudulent validations. Security CAA records reduce the attack surface of fraudulent validations by limiting which validation methods may be used and thus eliminating the risk posed by less-secure validation methods. Particularly, domains without a security CAA record are often highly vulnerable to man-in-the-middle adversaries that can intercept communications from a CA to the victim’s domain. The security Property significantly reduces this attack surface.

As with any restriction on certificate issuance, this introduces the potential for a Denial of Service (DoS) attack. There are two potential approaches to launching a DoS attack via security CAA records. The first is to attack a domain and spoof the existence of a security CAA record in order to prevent the domain owner from renewing his or her certificate (presuming the domain under attack was not using a validation method compliant with the security CAA record). This attack vector is not novel to security CAA records and is enabled solely by following the procedure specified in [RFC8659]. Per [RFC8659], the presence of any not-understood CAA record with the critical flag prevents issuance. Thus, the adoption of security CAA records does not increase the attack surface for this form of DoS attack as a gibberish CAA record with the critical flag set could lead to the same type of attack.

A second approach to a DoS attack enabled by security CAA records is to target a domain already using a security CAA record and interfere with all of the permitted validation methods with the idea that the presence of the security CAA will prevent the domain from falling back on alternative validation methods. This attack vector is mitigated by the diversity of different methods available to domain owners for validating domain ownership using security CAA records. A domain owner may use an alternate method to satisfy the security CAA record. In the event that a domain owner truly cannot satisfy any cryptographically-constrained domain validation method, the domain owner can still mitigate this attack by removing the security CAA record, obtaining a certificate, and then reinstating the security CAA record once the attack is over. As with all CAA records, CAs should not cache stale CAA record lookups that block issuance and should instead recheck the CAA record set when a new issuance request is received.

5. IANA Considerations

This document has no IANA actions.

6. 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/rfc/rfc2119>.

[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/rfc/rfc5234>.

[RFC7129] Gieben, R. and W. Mekking, "Authenticated Denial of Existence in the DNS", RFC 7129, DOI 10.17487/RFC7129, February 2014, <https://www.rfc-editor.org/rfc/rfc7129>.

[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/rfc/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/rfc/rfc8555>.

[RFC8657] Landau, H., "Certification Authority Authorization (CAA) Record Extensions for Account URI and Automatic Certificate Management Environment (ACME) Method Binding", RFC 8657, DOI 10.17487/RFC8657, November 2019, <https://www.rfc-editor.org/rfc/rfc8657>.

[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/rfc/rfc8659>.

[RFC9539] Gillmor, D. K., Ed., Salazar, J., Ed., and P. Hoffman, Ed., "Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS", RFC 9539, DOI 10.17487/RFC9539, February 2024, <https://www.rfc-editor.org/rfc/rfc9539>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Henry Birge-Lee
Princeton University
Email: birgelee@princeton.edu

Grace Cimaszewski
Princeton University
Email: gcimaszewski@princeton.edu

Cyrill E. Kr辰henb端hl
Princeton University
Email: cyrill.k@princeton.edu

Liang Wang
Princeton University
Email: lw19@princeton.edu

Aaron Gable
ISRG
Email: aaron@letsencrypt.org

Prateek Mittal
Princeton University
Email: pmittal@princeton.edu