Workgroup: Internet Engineering Task Force
Internet-Draft: draft-misell-acme-onion-00
Updates: RFC8555 (if approved)
Published: 24 February 2023
Intended Status: Standards Track
Expires: 28 August 2023
Authors: Q. Misell, Ed.
AS207960

Automated Certificate Management Environment (ACME) Extensions for ".onion" Domain Names

Abstract

The documents defines extensions to the Automated Certificate Management Environment (ACME) to allow for the automatic issuance of certificates to Tor hidden services (".onion" domains).

Discussion

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

Source for this draft and an issue tracker can be found at https://github.com/AS207960/acme-onion.

The project website and a reference implementation can be found at https://acmeforonions.org.

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 28 August 2023.

Copyright Notice

Table of Contents

1. Introduction

The Tor network has the ability to host "Onion Services" [tor-rend-spec-v3] [tor-address-spec] only accessible via the Tor network. These use the special use ".onion" top-level domain [RFC7686] to identify these services. These can be used as any other domain name could, but do not form part of the DNS infrastructure.

The Automated Certificate Management Environment (ACME) [RFC8555] defines challenges for validating control of DNS identifiers, and whilst a ".onion" domain may appear as a DNS name, it requires special consideration to validate control of one such that ACME could be used on ".onion" domains.

In order to allow ACME to be utilised to issue certificates to ".onion" domains this document specifies challenges suitable to validate control of these domains. Additionally this document defines an alternative to the DNS Certification Authority Authorization (CAA) Resource Record [RFC8659] that can be used with ".onion" domains.

1.1. Requirements Language

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 [BCP14] when, and only when, they appear in all capitals, as shown here.

2. Identifier

[RFC8555] defines the "dns" identifier type. This identifier type **MUST** be used when requesting a certificate for a ".onion" domain. The value of identifier **MUST** be the textual representation as defined in [tor-address-spec] §3. The value **MAY** include subdomain labels. Version 2 addresses **MUST NOT** be used as these are now considered insecure.

Example identifiers:

```
{
  "type": "dns",
  "value": "bbcweb3hytmzhn5d532owbu6oqadra5z3ar726vq5kgwwn6aucdccrad.oni
}

{
  "type": "dns",
  "value": "www.bbcweb3hytmzhn5d532owbu6oqadra5z3ar726vq5kgwwn6aucdccrad
}
```

3. Identifier Validation Challenges

The CA/Browser Forum Baseline Requirements [cabf-br] §B.2 define methods accepted by the CA industry for validation of ".onion" domains. This document incorporates these methods into ACME challenges.

3.1. Existing challenges

3.1.1. Existing "dns-01" Challenge

The existing "dns-01" challenge **MUST NOT** be used to validate ".onion" domains.

3.1.2. Existing "http-01" Challenge

The "http-01" challenge is defined as in [RFC8555] §8.3 may be used to validate a ".onion" domain, with the following modifications.

The ACME server **MUST** make its own connection to the hidden service via the Tor network, and **MUST NOT** outsource this, such as by using Tor2Web.

An additional field is defined to allow the ACME server to advertise the ed25519 public key it will use (as per [tor-rend-spec-v3] INTRO-AUTH) to authenticate itself during the introduction. This allows the hidden service to obtain a certificate,

authKey (optional, object) The Ed25519 public key encoded as per [RFC8037].

ACME servers **MUST NOT** use the same public key with multiple hidden services. ACME servers **MAY** re-use public keys for re-validation of the same hidden service.

The ACME server **SHOULD** follow redirects; note that these may be redirects to non ".onion" services, and the server **SHOULD** honour these.

3.1.3. Existing "tls-alpn-01" Challenge

The "tls-alpn-01" challenge is defined as in [RFC8737] may be used to validate a ".onion" domain, with the following modifications.

The ACME server **MUST** make its own connection to the hidden service via the Tor network, and **MUST NOT** outsource this, such as by using Tor2Web.

An additional field is defined to allow the ACME server to advertise the ed25519 public key it will use (as per [tor-rend-spec-v3] INTRO-AUTH) to authenticate itself during the introduction. This allows the hidden service to obtain a certificate,

authKey (optional, object) The Ed25519 public key encoded as per [RFC8037].

ACME servers **MUST NOT** use the same public key with multiple hidden services. ACME servers **MAY** re-use public keys for re-validation of the same hidden service.

3.2. New "onion-csr-01" Challenge

The two methods already defined in ACME and allowed by the CA/BF do not allow issuance of wildcard certificates. This new validation method incorporates the specially signed CSR (as defined by [cabf-br] §B.2.b) into ACME to allow for the issuance of wildcard certificates.

To this end a new challenge type called is defined, with the following fields:

type (required, string) The string "onion-csr-01"

nonce (required, string)
A Base64 [RFC4648] encoded nonce, including padding characters. It **MUST** contain at least 64 bits of entropy. It **MUST NOT** be valid for more than 30 days.

authKey (optional, object) The Ed25519 public key encoded as per [RFC8037].

```
{
  "type": "onion-csr-01",
  "url": "https://example.com/acme/chall/bbc625c5",
  "status": "pending",
  "nonce": "bI6/MRqV4gw=",
  "authKey": { ... }
}
```

An "authKey" field is defined to allow the ACME server to advertise the ed25519 public key it will use to decrypt the second layer descriptor to check CAA records.

ACME servers **MUST NOT** use the same public key with multiple hidden services. ACME servers **MAY** re-use public keys for re-validation of the same hidden service.

Clients may prove control over the key associated with the ".onion" service by generating their CSR with the following additional attributes and signing it with the private key of the ".onion" domain:

*A caSigningNonce attribute containing the nonce provided in the challenge. This **MUST NOT** be base64 encoded in the CSR.

*An applicantSigningNonce containing a nonce generated by the client. This **MUST** have at least 64 bits of entropy.

These additional attributes have the following format

```
id-pkix  OBJECT IDENTIFIER ::=
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) }

id-pe OBJECT IDENTIFIER ::= { id-pkix 1 }

cabf OBJECT IDENTIFIER ::=
  { joint-iso-itu-t(2) international-organizations(23)
    ca-browser-forum(140) }

cabf-caSigningNonce OBJECT IDENTIFIER ::= { cabf 41 }

caSigningNonce ATTRIBUTE ::= {
  WITH SYNTAX             OCTET STRING
  EQUALITY MATCHING RULE  octetStringMatch
  SINGLE VALUE            TRUE
  ID                      { cabf-caSigningNonce }
}

cabf-applicantSigningNonce OBJECT IDENTIFIER ::= { cabf 42 }

applicantSigningNonce ATTRIBUTE ::= {
  WITH SYNTAX             OCTET STRING
  EQUALITY MATCHING RULE  octetStringMatch
  SINGLE VALUE            TRUE
  ID                      { cabf-applicantSigningNonce }
}
```

In a variation to the usual state machine of ACME, a client need not respond to the challenge. The act of POSTing a CSR to the finalization endpoint is in itself a response to the challenge. The challenge and order progress directly to either the "valid" or "invalid" state without passing through "processing" or "ready" (respectively).

In the case of a CSR posted to the finalization endpoint that does not include the above extensions the order **SHOULD** remain in the "pending" state and **SHOULD NOT** transition to the "invalid" state. Only in the case that an "onion-csr-01" CSR is POSTed to the finalization endpoint that subsequently fails validation **MUST** the order transition to the "invalid" state.

4. Certification Authority Authorization (CAA)

".onion" domains are not part of the DNS, and as such a variation on CAA [RFC8659] is required to allow restrictions to be placed on certificate issuance.

To this end a new field is added to the second layer hidden service descriptor [tor-rend-spec-v3] § 2.5.2.2. with the following format:

```
"caa" SP flags SP tag SP value NL
[Any number of times]
```

The contents of "flag", "tag", and "value" are as per [RFC8659] § 4.1.1. Multiple CAA records may be present, as is the case in the DNS. CAA records in a hidden service descriptor are to be treated the same by CAs as if they had been at the DNS for the ".onion" domain.

A hidden service's second layer descriptor using CAA may look something like the following:

```
create2-formats 2
single-onion-service
caa 0 issue "example.com"
caa 0 iodef "mailto:security@example.com"
caa 128 validationmethods "onion-csr-01"
introduction-point AwAGsAk5nSMpAhRqhMHbTFCTSlfhP8f5PqUhe6DatgMgk7kSL3KHC
...
```

4.1. Relevant Resource Record Set

In the absence of the possibility of delegation from a ".onion" domain as there is in the DNS there is no need, nor indeed any possibility to search up a the DNS tree for a relevant CAA record set. Instead all subdomains under a ".onion" domain share the same CAA record set. That is all of these share a CAA record set with "a.onion":

- *b.a.onion
- *c.a.onion
- *e.d.a.onion

But these do not:

- *b.c.onion
- *c.d.onion
- *e.c.d.onion

4.2. When to check CAA

If the hidden service has client authentication enabled then it will be impossible for the CAA to decrypt the second layer descriptor to read the CAA records until the CAAs public key has been added to first layer descriptor. To this end a CA **SHOULD** wait until the client responds to an authorization, and treat this as indication that their public key has been added and that the CA will be able to decrypt the second layer descriptor.

4.3. Preventing mis-issuance by unknown CAs

As the CAA records are in the second layer descriptor and in the case of a hidden service requiring client authentication it is

impossible to read them without the hidden service trusting a CAs public key, a method is required to signal that there are CAA records present (but not reveal their contents, which may disclose unwanted information about the hidden service operator).

To this end a new field is added to the first layer hidden service descriptor [tor-rend-spec-v3] § 2.5.1.2. with the following format:

"caa-critical" NL
[At most once]

If a CA encounters this flag it **MUST NOT** proceed with issuance until it can decrypt and parse the CAA records from the second layer descriptor.

5. IANA Considerations

5.1. Validation Methods

Per this document, one new entry has been added to the "ACME Validation Methods" registry defined in [RFC8555] §9.7.8. This entry is defined below:

Label	Identifier Type	ACME	Reference
onion-csr-01	dns	Y	draft-misell-acme-onion-latest

Table 1: New entries

6. Security Considerations

6.1. Use of "dns" identifier type

The re-use of the "dns" identifier type for a domain not actually in the DNS infrastructure raises questions regarding its suitability. The reasons the author wish to pursue this path in the first place are detailed in Appendix A. It is felt that there is little security concern in reuse of the "dns" identifier type with regards the mis-issuance by CAs that are not aware of ".onion" domains.

6.1.1. "http-01" Challenge

The CA would follow the procedure set out in [RFC8555] §8.3 which specifies that the CA should "Dereference the URL using an HTTP GET request". Given that ".onion" require special handling to dereference, this de-referencing will fail, disallowing issuance.

6.1.2. "tls-alpn-01" Challenge

The CA would follow the procedure set out in [RFC8737] §3 which specifies that the CA "resolves the domain name being validated and chooses one of the IP addresses returned for validation". Given that ".onion" are not resolvable to IP addresses, this de-referencing will fail, disallowing issuance.

6.1.3. "dns-01" Challenge

The CA would follow the procedure set out in [RFC8555] §8.4 which specifies that the CA should "query for TXT records for the validation domain name". Given that ".onion" are not present in the DNS infrastructure, this query will fail, disallowing issuance.

6.2. Key Authorization with "onion-csr-01"

The "onion-csr-01" challenge does not make use of the key authorization string defined in [RFC8555] §8.1. This does not weaken the integrity of authorizations.

The key authorization exists to ensure that an attacker observing the validation channel can observe the correct validation response, but cannot compromise the integrity of authorizations as the response only be used with the account key for which it was generated. As the validation channel for this challenge is ACME itself, and ACME already requires that the request be signed by the account, the key authorization is not required.

6.3. Use of Tor for non ".onion" domains

An ACME server **MUST NOT** utilise Tor for the validation of non ".onion" domains, due to the risk of possible exit hijacking.

6.4. Security of CAA records

The second layer descriptor is encrypted and MACed in a way that only a party with access to the secret key of the hidden service could manipulate what is published there. For more information about this process see [tor-rend-spec-v3] § 2.5.3.

7. References

7.1. Normative References

[BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017. <https://www.rfc-editor.org/info/bcp14>

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use Domain Name", RFC 7686, DOI 10.17487/RFC7686, October 2015, <https://www.rfc-editor.org/info/rfc7686>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption

(JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/info/rfc8037>.

[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>.

[RFC8737] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) TLS Application-Layer Protocol Negotiation (ALPN) Challenge Extension", RFC 8737, DOI 10.17487/RFC8737, February 2020, <https://www.rfc-editor.org/info/rfc8737>.

[tor-address-spec] Nick Mathewson, N., "Special Hostnames in Tor", <https://spec.torproject.org/address-spec>.

[tor-rend-spec-v3] The Tor Project, "Tor Rendezvous Specification - Version 3", <https://spec.torproject.org/rend-spec-v3>.

[cabf-br] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.8.6.pdf>.

7.2. Informative References

[onion-services-setup] The Tor Project, "Set Up Your Onion Service", <https://community.torproject.org/onion-services/setup/>.

Appendix A. Discussion on the use of the "dns" identifier type

The reasons for utilising the "dns" identifier type in ACME and not defining a new identifier type for ".onion" domains may not seem obvious at first glance. After all, ".onion" domains are not part of the DNS infrastructure and as such why should they use the "dns" identifier type?

The CA/Browser Forum Baseline Requirements [cabf-br] §B.2.a.ii define, and this standard allows, using the "http-01" or "tls-alpn-01" validation methods already present in ACME (with some considerations). Given the situation of a web server placed behind a Tor terminating proxy (as per the setup suggested by the Tor project [onion-services-setup]), existing ACME tooling can be blind to the fact that a ".onion" domain is being utilised, as they simply receive an incoming TCP connection as they would regardless (albeit from the Tor terminating proxy).

An example of this would be Certbot placing the ACME challenge response file in the webroot of an NGINX web server. Neither Certbot nor NGINX would require any modification to be aware of any special handling for ".onion" domains.

This does raise some questions regarding security within existing implementations, however the authors believe this is of little concern, as per Section 6.1.

Acknowledgements

With thanks to the Open Technology Fund for funding the work that went into this document.

The authors also wish to thank the following for their input on this document:

*Iain R. Learmonth

Author's Address

Q Misell (editor)
AS207960 Cyfyngedig
13 Pen-y-lan Terrace
Caerdydd
CF23 9EU
United Kingdom

Email: q@magicalcodewit.ch, q@as207970.net
URI: https://magicalcodewit.ch