Workgroup:	ACME Working Group
Internet-Draft:	draft-suchan-acme-onionv3-00
Published:	2 May 2022
Intended Status:	Standards Track
Expires:	3 November 2022
Author:	Seo Suchan

Automated Certificate Management Environment (ACME) Onion Identifier Validation Extension

Abstract

This document specifies identifiers and challenges required to enable the Automated Certificate Management Environment (ACME) to issue certificates for Tor Project's onion V3 addresses.

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 3 November 2022.

Copyright Notice

Table of Contents

1. Introduction
2. Terminology
3. Onion Identifier
4. Validation Challenges for onion address
   4.1. CSR signed with Onion public key challagne
5. IANA Considerations
   5.1. Identifier Types
   5.2. Challenge Types
6. Security Considerations
7. note
8. Normative References

Author's Address

1. Introduction

While onion addresses are in form of DNS address, they aren't in part of ICANN hierarchy, and onion name's self-verifying construction warrents different verification, duce different identfier type for them is worth consider.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, [RFC2119].

3. Onion Identifier

[RFC8555] only defines the identifier type "dns", but it assumes it's on public CA/B hirearuchy

An identifier for the onion address acmeulkebl5k66q4sgcgzpe3tddrffnpn5gpnkpamz4zcuv5hk7fqwad.onion would be formatted like:

```
{"type": "onion-v3", "value":
"acmeulkebl5k66q4sgcgzpe3tddrffnpn5gpnkpamz4zcuv5hk7fqwad.onion"}
```

4. Validation Challenges for onion address

Onion-v3 identifiers MAY be used with the existing "http-01" and "tls-alpn-01" challenges from [RFC8555] Section 8.3 and [RFC8737] Section 3 respectively. To use Onion identifiers with these challenges their initial DNS resolution step MUST be skipped and the approperate Tor daemon that in control of CA MUST used to proxy such request.

The exsisting "dns-01" challange MUST NOT be used to validate onion addresses.

In addition to challanges earlier RFC defined, there

4.1. CSR signed with Onion public key challagne

With Onion-csr validation, the client in an ACME transaction proves its control of oninon address by proving the possesion of onion hidden service identity key. The ACME server challenges the client to sign CSR that includes the nonce it gave with.

The Onion-csr ACME challenge object has the following format:

type (required, string): The string "onion-v3-csr

token (required, string): A random value that uniquely identifies the challenge. This value MUST have at least 128 bits of entropy. It MUST NOT contain any characters outside the base64url alphabet as described in [RFC4648] Section 5. Trailing '=' padding characters MUST be stripped. See [RFC4086] for additional information on randomness requirements.

The client prepares for validation by constructing a self-signed CSR that MUST contain an cabf caSigningNonce Attribute and a subjectAlternativeName extension [RFC5280]. The subjectAlternativeName extension MUST contain a single dNSName entry where the value is the domain name being validated. The cabf caSigningNonce Attribute MUST contain the token string as ascii encoded for the challenge.

The cabf caSigningNonce Attribute is identified by the cabf-caSigningNonce object identifier (OID) in the cabf arc [RFC5280]. conseurt [cabr] appendex B for how to construct CSR itself in detail.

A client fulfills this challenge by construct the challange CSR from the "token" value provided in the challange, then POST on challange URL with crafted CSR as payload to request validated by the server.

```
POST /acme/authz/1234/1
Host: example.com
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/1",
    "nonce": "XaYcb5XoTUgRWbTWw_NwkcP",
    "url": "https://example.com/acme/authz/1234/1"
  }),
  "payload": base64url({
       "csr": "MIIBBzCBugIBADAAMCo...gRYTMAhRP8nIH",
     }),
  "signature": "0wSzBJBgNVHREEQ...gYJKoZIhvcNAQkOM"
}
```

On receiving this request from client, the server verifies client's control over the onion address by verifiy that CSR is crafted with expected properties:

1. CSR is signed with private part of identity key the requested onion address made from.
2. A caSigningNonce attribute that contains token Value that challenge object provided.

5. IANA Considerations

5.1. Identifier Types

Adds a new type to the Identifier list defined in Section 9.7.7 of [RFC8555] with the label "onion-v3" and reference I-D.ietf-acme-onion.

5.2. Challenge Types

in the Validation Methods list defined in Section 9.7.8 of [RFC8555]:

Adds the raw "onin-challange-csr" to the Validation Methods.

Adds the value "onion-v3-csr" to the Identifier Type column for the "http-01", "onion-challange-csr", and "tls-alpn-01" challenges.

6. Security Considerations

As onion addresses are able to generated in massive quantity without financial cost, it bypasses the normal ratelimit CAs imposess. CAs SHOULD adapt some mesure to prevent DoSing the CA by create hugh amount of request for onion address. For exemple, imposing limit per ACME account or require order to have at least one non-onion domain.

7. note

this doc is made as documentation of my pebble tree does: process may change in track

8. Normative References

[cabr] Forum, C., "CA/B forum baseline requirement", 2022, <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.8.4.pdf>.

[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>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[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>.

[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>.

[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>.

[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>.

Author's Address

Seo Suchan
Email: tjtncks@gmail.com