Workgroup: ACME Working Group
Internet-Draft: draft-shoemaker-acme-onion-01
Published: 7 July 2020
Intended Status: Standards Track
Expires: 8 January 2021
Authors: R.B. Shoemaker
ISRG

Automatic Certificate Management Environment (ACME) Onion v3 Identifier Validation Extension

Abstract

This document specifies identifiers and challenges required to enable the Automatic Certificate Management Environment (ACME) to issue certificates for Onion Addresses as specified in Tor Rendezvous Specification - Version 3.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Currently the Automatic Certificate Management Environment (ACME) [RFC8555] only specifies how DNS identifiers and IP address identifiers [RFC8738] may be validated for inclusion in x.509 certificates [RFC5280]. This document extends the protocol to include a validation mechanism for Tor version 3 Onion Addresses [TOR-REND-V3].

2. Terminology

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.

3. Onion Address Identifier

Version 3 Onion address identifier objects MUST use the type "onion-v3". The value field of the identifier MUST contain the textual encoding of the address as defined as onion_address in [TOR-REND-V3] section 6. ACME servers MUST verify that the value field contains a properly encoded address by checking that it contains only two labels, that first label contains a valid checksum, and that the last byte of the first label is \x03. [TODO: this could probably be specified better?]

An identifier for the version 3 Onion address address for the following Ed25519 public key would be formatted like so:

```
-----BEGIN PUBLIC KEY-----
MCowBQYDK2VwAyEAAJhLnbvXNWu8WXre3Y0HU+1FErU13zcbO7pEqkI38+Q=
-----END PUBLIC KEY-----
```

```
{"type": "onion-v3", "value": "acmexhn3242wxpczplpn3dihkpwukevvgxptogz3x
```

4. Onion CSR Challenge

This document specifies a single new challenge type that can be used for validation of onion-v3 identifiers. This challenge demonstrates control of the private key associated with the public key contained within the Onion address by signing a CSR containing special attributes. The challenge object contains the following fields:

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

nonce (required, string): A random value that uniquely identifies the challenge. This value MUST have at least 64 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 Certificate Signing Request (CSR) [RFC2986]. This CSR MUST contain two attributes, a caSigningNonce attribute containing the nonce provided in the challenge object, and a applicantSigningNonce attribute containing a random value picked by the client. This random value MUST have at least 64 bits of entropy. The CSR MUST be signed by the private key associated with the public key contained within the Onion address.

The caSigningNonce and applicantSigningNonce attributes are defined as follows in [CABF-BR] Appendix F

```
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 }
}
```

The client completes the challenge process by POSTing a JSON object containing the signed CSR they generated to the challenge URL. The base64url encoding of the protected headers and payload is described in Section 6.1 of [RFC8555]. The JSON object contains the following fields:

csr (required, string): The base64url-encoded DER encoding of the signed CSR.

On receiving this request from a client the ACME server verifies the CSR by checking that it contains the caSigningNonce attribute, and that it's value matches the nonce in the challenge object it created, the applicantSigningNonce, and that the value contains a random value with at least 64 bits of entropy, and that the signature can be verified using the public key encoded in the Onion address that is being validated. If all of these checks succeed, then the validation is successful. Otherwise, it is a failure.

5. IANA Considerations

5.1. Identifier Types

Adds a new type to the "ACME Identifier Types" registry defined in Section 9.7.7 of [RFC8555] with Label "onion-v3" and Reference "I-D.shoemaker-acme-onion".

5.2. Challenge Types

Adds one new entry to the "ACME Validation Methods" registry defined in Section 9.7.8 of [RFC8555] as defined below.

Label	Identifier Type	ACME	Reference
onion-v3-csr	onion-v3	Y	I-D.shoemaker-acme-onion

Table 1

6. Security Considerations

[NOTE: Probably should consider *something* here (may want to reference [TOR-REND-V3] Section 2.2.7?).]

7. Acknowledgments

The author would like to thank those who offered editorial and technical input on the document. Special thanks to the participants in the CA/Browser who specified the initial validation mechanisms and controls for Onion Addresses.

8. Normative References

[TOR-REND-V3]

Tor Project, "Tor Rendezvous Specification - Version 3", 2020, <https://spec.torproject.org/rend-spec-v3>.

[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>.

[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>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[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>.

[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>.

[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>.

[CABF-BR] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, Version 1.6.8", 2020, <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.6.8.pdf>.

[RFC8738] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) IP Identifier Validation Extension", RFC 8738, DOI 10.17487/RFC8738, February 2020, <https://www.rfc-editor.org/info/rfc8738>.

9. Informative References

[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>.

Author's Address

Roland Bracewell Shoemaker
Internet Security Research Group

Email: roland@letsencrypt.org