Workgroup: Network Working Group
Internet-Draft:
draft-ietf-dnsop-domain-verification-
techniques-04
Published: 3 March 2024
Intended Status: Best Current Practice
Expires: 4 September 2024
Authors: S. Sahib
Brave Software
S. Huque
Salesforce
P. Wouters
Aiven
E. Nygren
Akamai Technologies

Domain Control Validation using DNS

Abstract

Many application services on the Internet need to verify ownership or control of a domain in the Domain Name System (DNS). The general term for this process is "Domain Control Validation", and can be done using a variety of methods such as email, HTTP/HTTPS, or the DNS itself. This document focuses only on DNS-based methods, which typically involve the application service provider requesting a DNS record with a specific format and content to be visible in the requester's domain. There is wide variation in the details of these methods today. This document proposes some best practices to avoid known problems.

Discussion Venues

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/ietf-wg-dnsop/draft-ietf-dnsop-domain-verification-techniques/.

Status of This Memo

This Internet-Draft will expire on 4 September 2024.

Table of Contents

1. Introduction

Many providers of internet services need domain owners to prove that they control a particular DNS domain before the provider can operate services for or grant some privilege to that domain. For instance, Certification Authorities (CAs) ask requesters of TLS certificates to prove that they operate the domain they are requesting the certificate for. Providers generally allow for several different ways of proving control of a domain. In practice, DNS-based methods take the form of the provider generating a random token and asking the requester to create a DNS record containing this random token and placing it at a location within the domain that the provider can query for. Generally only one temporary DNS record is sufficient for proving domain ownership, although sometimes the DNS record must be kept in the zone to prove continued ownership of the domain.

This document describes pitfalls associated with some common practices using DNS-based techniques deployed today, and recommends using TXT based domain control validation in a way that is time-bound and targeted to the service. The Appendix A includes a more detailed survey of different methods used by a set of application service providers.

Other techniques such as email or HTTP(S) based validation are out-of-scope.

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.

* Validation record: the DNS record that is used to prove ownership of a domain name ([RFC8499]). It typically contains an unguessable value generated by the provider which serves as a challenge. The provider looks for the validation record in the zone of the domain name being verified and checks if it contains the unguessable value.

* Provider: an internet-based provider of a service, for e.g., a Certification Authority or a service that allows for user-controlled websites. These services often require a user to verify that they control a domain. The provider may be implementing a standard protocol for domain validation (such as [RFC8555]) or they may have their own specification.

* Intermediary: an internet-based service that leverages the services of other providers on behalf of a user. For example, an

intermediary might be a service that allows for user-controlled websites and in-turn needs to use a Certification Authority provider to get TLS certificates for the user on behalf of the website.

*User: the owner or operator of a domain in the DNS who needs to prove ownership of that domain to a provider.

*Random Token: a random value that uniquely identifies the DNS domain control validation challenge, defined in Section 5.1.3.

3. Common Pitfalls

A very common but unfortunate technique in use today is to employ a DNS TXT record and placing it at the exact domain name whose control is being validated. This has a number of known operational issues. If the domain owner uses multiple application services using this technique, it will end up deploying a DNS TXT record "set" at the domain name, containing one TXT record for each of the services.

Since DNS resource record sets are treated atomically, a query for the validation record will return all TXT records in the response. There is no way for the verifier to specifically query only the TXT record that is pertinent to their application service. The verifier must obtain the aggregate response and search through it to find the specific record it is interested in.

Additionally, placing many such TXT records at the same name increases the size of the DNS response. If the size of the UDP response (UDP being the most common DNS transport today) is large enough that it does not fit into the Path MTU of the network path, this may result in IP fragmentation, which often does not work reliably on the Internet today due to firewalls and middleboxes, and also is vulnerable to various attacks ([AVOID-FRAGMENTATION]). Depending on message size limits configured or being negotiated, it may alternatively cause the DNS server to "truncate" the UDP response and force the DNS client to re-try the query over TCP in order to get the full response. Not all networks properly transport DNS over TCP and some DNS software mistakenly believe TCP support is optional ([RFC9210]).

Other possible issues may occur. If a TXT record (or any other record type) is designed to be placed at the same domain name that is being validated, it may not be possible to do so if that name already has a CNAME record. This is because CNAME records cannot co-exist with other records at the same name. This situation cannot occur at the apex of a DNS zone, but can at a name deeper within the zone.

When multiple distinct services create domain validation records at the same domain name, there is no way to delegate an application specific domain validation record to a third party. Furthermore, even without delegation, an organization may have a shared DNS zone where they need to provide record level permissions to the specific division within the organization that is responsible for the application in question. This can't be done if all applications share the domain name.

This specification proposes the use of application-specific labels in the domain validation record to address these issues.

4. Scope of Validation

For security reasons, it is crucial to understand the scope of the domain name being validated. Both application service providers and the domain owner need to clearly specify and understand whether the validation request is for a single hostname, a wildcard (all hostnames immediately under that domain), or for the entire domain and subdomains rooted at that name. This is particularly important in large multi-tenant enterprises, where an individual deployer of a service may not necessarily have operational authority of an entire domain.

In the case of X.509 certificate issuance, the certificate signing request and associated challenge are clear about whether they are for a single host or a wildcard domain. Unfortunately, the ACME protocol's DNS-01 challenge mechanism ([RFC8555], Section 8.4) does not differentiate these cases in the DNS validation record. In the absence of this distinction, the DNS administrator tasked with deploying the validation record may need to explicitly confirm the details of the certificate issuance request to make sure the certificate is not given broader authority than the domain owner intended.

In the more general case of an Internet application service granting authority to a domain owner, again no existing DNS challenge scheme makes this distinction today. New applications should consider having different application names for different scopes, as described below in Section 5.1.2. Regardless, services should very clearly indicate the scope of the validation in their public documentation so that the domain administrator can use this information to assess whether the validation record is granting the appropriately scoped authority.

4.1. Domain Boundaries

The hierarchical structure of domain names do not necessarily define boundaries of ownership and administrative control (e.g., as

discussed in [I-D.draft-tjw-dbound2-problem-statement]). Some domain names are "public suffixes" ([RFC8499]) where care may need to be taken when validating control. For example, there are security risks if a provider can be tricked into believing that an attacker has control over ".co.uk" or ".com". The volunteer-managed Public Suffix List [PSL] is one mechanism available today that can be useful for identifying public suffixes.

Future specifications may provide better mechanisms or recommendations for defining domain boundaries or for enabling organizational administrators to place constraints on domains and subdomains. See Appendix A.1.2.4 for cases where DNS records can be used as constraints complementary to domain verification.

5. Recommendations

5.1. Validation Record Format

5.1.1. Name

The RECOMMENDED format is application-specific underscore prefix labels. Domain Control Validation records are constructed by the provider by prepending the label "_<PROVIDER_RELEVANT_NAME>-challenge" to the domain name being validated (e.g. "_foo-challenge.example.com"). The prefixed "_" is used to avoid collisions with existing hostnames.

5.1.2. Scope Indication

For applications that may apply more broadly than to a single host name, the RECOMMENDED approach is to differentiate the application-specific underscore prefix labels to also include the scope (see #scope). In particular:

* "_<PROVIDER_RELEVANT_NAME>-host-challenge.example.com" applies only to the specific host name of "example.com" and not to anything underneath it.

* "_<PROVIDER_RELEVANT_NAME>-wildcard-challenge.example.com" applies to all host names at the level immediately underneath "example.com". For example, it would apply to "foo.example.com" but not "example.com" nor "quux.bar.example.com"

* "_<PROVIDER_RELEVANT_NAME>-domain-challenge.example.com" applies to the entire domain "example.com" as well as its subdomains. For example, it would apply to all of "example.com", "foo.example.com", and "quux.bar.example.com"

The application provider will normally know which of these scoped DNS records to query based on the user's requested configuration. So

this does not typically result in multiple queries for different possible scopes. If discovery of scope is needed for a specific application as part of the domain control validation process, then the scope could alternatively be encoded in a key value pair in the record data.

Note that a proposed update to the ACME DNS challenge specification [ACME-SCOPED-CHALLENGE] has incorporated this scope indication format.

5.1.3. Random Token

A unique token used in the challenge. It should be a random value issued between parties (provider to user, provider to intermediary, or intermediary to user) with the following properties:

1. MUST have at least 128 bits of entropy.

2. base64url ([RFC4648], Section 5) encoded, base32 ([RFC4648], Section 6) encoded, or base16 ([RFC4648], Section 8) encoded.

See [RFC4086] for additional information on randomness requirements.

Base32 encoding or hexadecimal base16 encoding are RECOMMENDED to be specified when the random token would exist in a DNS label such as in a CNAME target. This is because base64 relies mixed case (and DNS is case-insensitive as clarified in [RFC4343]) and because some base64 characters ("/", "+", and "=") may not be permitted by implementations that limit allowed characters to those allowed in hostnames. If base32 is used, it SHOULD be specified in way that safely omits the trailing padding ("="). Note that DNS labels are limited to 63 octets which limits how large such a token may be.

This random token is placed in the RDATA as described in the rest of this section.

5.2. TXT Record

The RECOMMENDED method of doing DNS-based domain control validation is to use DNS TXT records. The name is constructed as described in Section 5.1.1, and RDATA MUST contain at least a Random Token (constructed as in Section 5.1.3). If metadata (see Section 5.2.1) is not used, then the unique token generated as-above can be placed as the only contents of the RDATA. For example:

```
_foo-challenge.example.com.  IN   TXT  "3419...3d206c4"
```

If a provider has an application-specific need to have multiple validations for the same label, multiple prefixes can be used:

```
_feature1._foo-challenge.example.com.  IN   TXT  "3419...3d206c4"
```

This again allows the provider to query only for application-specific records it needs, while giving flexibility to the user adding the DNS record (i.e. they can be given permission to only add records under a specific prefix by the DNS administrator). Whether or not multiple validation records can exist for the same domain is up to the provider's application specification.

Consumers of the provider services need to relay information from a provider's website or APIs to their local DNS administrators. The exact DNS record type, content and location is often not clear when the DNS administrator receives the information, especially to consumers who are not DNS experts. Providers SHOULD offer detailed help pages, that are accessible without needing a login on the provider website, as the DNS administrator often has no login account on the provider service website. Similarly, for clarity, the exact and full DNS record (including a Fully Qualified Domain Name) to be added SHOULD be provided along with help instructions.

Providers MUST validate that a random token in the TXT record matches the one that they gave to the user for that specific domain name.

5.2.1. Metadata For Expiry

Providers MUST provide clear instructions on when a validation record can be removed. These instructions SHOULD be encoded in the RDATA via comma-separated ASCII key-value pairs [RFC1464], using the key "expiry" to hold a time after which it is safe to remove the validation record. If this key-value format is used, the verification token should use the key "token". For example:

```
_foo-challenge.example.com.  IN   TXT  "token=3419...3d206c4,expiry=2023
```

When a expiry time is specified, the value of "expiry" SHALL be in ISO 8601 format as specified in [RFC3339], Section 5.6.

Alternatively, if the record should never expire (for instance, if it may be checked periodically by the provider) and should not be removed, the key "expiry" SHALL be set to have value "never".

```
_foo-challenge.example.com.  IN   TXT  "token=3419...3d206c4,expiry=neve
```

The "expiry" key MAY be omitted in cases where the provider has clarified the record expiry policy out-of-band (Appendix A.1.1.3).

```
_foo-challenge.example.com.  IN   TXT  "token=3419...3d206c4"
```

Note that this is semantically the same as:

```
_foo-challenge.example.com.  IN   TXT  "3419...3d206c4"
```

The user SHOULD de-provision the resource record provisioned for DNS-based domain control validation once it is no longer required.

5.3. CNAME Records

CNAME records MAY be used instead of TXT records, either for Delegated Domain Control Validation (Section 5.3.2) or where specified by providers to support users who are unable to create TXT records.

A provider supporting CNAME records MUST specify the use of an underscore-prefixed label (e.g., _foo-<token> or even the less-recommended _<token>) as a CNAME MUST NOT be placed at the same domain name that is being validated. This is for the same reason already cited in Section 3. CNAME records cannot co-exist with other data, and there may already be other record types that exist at the domain name. Instead, as with the TXT record recommendation, a provider specific label should be added as a subdomain of the domain to be verified. This ensures that the CNAME does not collide with other record types.

In practice, many providers that employ CNAMEs for domain control validation today use a random subdomain label, which also works to avoid collisions. But adding an provider-specific component in addition (such as _foo-<RANDOM>-challenge) would make it easier for the domain owner to keep track of why and for what service a validation record has been deployed.

Note that some DNS implementations permit the deployment of CNAME records co-existing with other record types. These implementations are in violation of the DNS protocol. Furthermore, they can cause resolution failures in unpredictable ways depending on the behavior of DNS resolvers, the order in which query types for the name are processed etc. In short, they cannot work reliably and these implementations should be fixed.

5.3.1. CNAME Records for Domain Control Validation

A provider may specify using CNAME records instead of TXT records for Domain Control Validation. In this case, the target of the CNAME would contain the base16-encoded (or base32-encoded) random token followed by a suffix specified by the provider. For example:

```
_foo-challenge.example.com.  IN   CNAME <random-token>.dcv.provider.exam
```

5.3.2. Delegated Domain Control Validation

Separately, CNAME records also enable delegated domain control validation, which lets the user delegate the domain control validation process for their domain to an intermediary without having to hand over full DNS access. The intermediary gives the user a CNAME record to add for the domain and provider being validated that points to the intermediary's DNS, where the actual validation TXT record is placed. The record name and base16-encoded (or base32-encoded) random tokens are generated as in Section 5.1. For example:

```
_foo-challenge.example.com.  IN   CNAME  "<intermediary-random-token>.dc
```

The intermediary then adds the actual validation record in a domain they control:

```
<intermediary-random-token>.dcv.intermediary.example. TXT "<provider-ran
```

Such a setup is especially useful when the provider wants to periodically re-issue the challenge. CNAMEs allow automating the renewal process by letting the intermediary place the random token in their DNS instead of needing continuous write access to the user's DNS.

Importantly, the CNAME record target also contains a random token issued by the intermediary to the user (preferably over a secure channel) which proves to the intermediary that example.com is controlled by the user. The intermediary must keep an association of users and domain names to the associated intermediary-random-tokens. Without a linkage validated by the intermediary during provisioning and renewal there is the risk that an attacker could leverage a "dangling CNAME" to perform a "subdomain takeover" attack ([SUBDOMAIN-TAKEOVER]).

When a user stops using the intermediary they should remove the domain control validation CNAME in addition to any other records they have associated with the intermediary.

See Appendix A.1.2.2 for examples.

5.3.3. Domain Control Validation Supporting Multiple Intermediaries

There are use-cases where a user may wish to simultaneously use multiple intermediaries or multiple independent accounts with a provider. For example, a hostname may be using a "multi-CDN" where the hostname simultaneously uses multiple Content Delivery Network (CDN) providers.

To support this, providers may support prefixing the challenge with a label containing an unique account identifier of the form

_<identifier-token> and following the requirements of Section 5.1.3, specified as either base32 or base16 encoded. This identifier token should be stable over time and would be provided to the user by the provider, or by an intermediary in the case where domain validation is delegated (Section 5.3.2).

The resulting record could either directly contain a TXT record or a CNAME (as in Section 5.3.2). For example:

```
_<identifier-token>._foo-challenge.example.com.  IN   TXT  "3419...3d206
```

or

```
_<identifier-token>._foo-challenge.example.com.  IN   CNAME  "<intermedi
```

When performing validation, the provider would resolve the DNS name containing the appropriate identifier token.

5.4. Time-bound checking

After domain control validation is completed, there is typically no need for the TXT or CNAME record to continue to exist as the presence of the domain validation DNS record for a service only implies that a user with access to the service also has DNS control of the domain at the time the code was generated. It should be safe to remove the validation DNS record once the validation is done and the service provider doing the validation should specify how long the validation will take (i.e. after how much time can the validation DNS record be deleted).

One exception is if the record is being used as part of a delegated domain control validation setup (Section 5.3.2); in that case, the CNAME record that points to the actual validation TXT record cannot be removed as long as the user is still relying on the intermediary.

5.5. DNAME

Domain control validation in the presence of a DNAME [RFC6672] is theoretically possible. Since a DNAME record redirects the entire subtree of names underneath the owner of the DNAME, it is not possible to place a validation record under the DNAME owner itself. It would have to be placed under the DNAME target name, since any lookups for a name under the DNAME owner will be redirected to the corresponding name under the DNAME target.

6. Security Considerations

A malicious service that promises to deliver something after domain control validation could surreptitiously ask another service provider to start processing or sending mail for the target domain

and then present the victim domain administrator with this DNS TXT record pretending to be for their service. Once the administrator has added the DNS TXT record, instead of getting their service, their domain is now certifying another service of which they are not aware they are now a consumer. If services use a clear description and name attribution in the required DNS TXT record, this can be avoided. For example, by requiring a DNS TXT record at _vendorname.example.com instead of at example.com, a malicious service could no longer replay this without the DNS administrator noticing this. Both the provider and the service being authenticated and authorized should be unambiguous from the TXT record owner name and RDATA content to prevent malicious services from misleading the domain owner into certifying a different provider or service.

Amiguity of scope introduces risks, as described in Section 4. Distinguishing the scope in the application-specific label, along with good documentation, should help make it clear to DNS administrators whether the record applies to a single host name, a wildcard, or an entire domain. Always using this indication rather than having a default scope reduces ambiguity, especially for protocols that may have used a shared application-specific label for different scopes in the past. While it would also have been possible to include the scope in as an attribute in the TXT record, that has more potential for ambiguity and misleading an operator, such as if an implementation ignores attribute it doesn't recognize but an attacker includes the attribute to mislead the DNS administrator.

Providers and intermediaries should use authenticated channels to convey instructions and random tokens to users. Otherwise an attacker in the middle could alter the instructions, potentially allowing the attacker to provision the service instead of the user.

A domain owner SHOULD sign their DNS zone using DNSSEC [RFC9364] to protect validation records against DNS spoofing attacks.

DNSSEC validation SHOULD be performed by service providers that verify validation records they have requested to be deployed. If no DNSSEC support is detected for the domain owner zone, or if DNSSEC validation cannot be performed, service providers SHOULD attempt to query and confirm the validation record by matching responses from multiple DNS resolvers on unpredictable geographically diverse IP addresses to reduce an attacker's ability to complete a challenge by spoofing DNS. Alternatively, service providers MAY perform multiple queries spread out over a longer time period to reduce the chance of receiving spoofed DNS answers.

6.1. Public Suffixes

As discussed above in Section 4.1, there are risks in allowing control to be demonstrated over domains which are "public suffixes" (such as ".co.uk" or ".com"). The volunteer-managed Public Suffix List ([PSL]) is one mechanism that can be used. It includes two "divisions" ([PSL-DIVISIONS]) covering both registry-owned public suffixes (the "ICANN" division) and a "PRIVATE" division covering domains submitted by the domain owner.

Operators of public suffix domains which are in the "PRIVATE" division often provide multi-tenant services such as dynamic DNS, web hosting, and CDN services. As such, they sometimes allow their sub-tenants to provision names as subdomains of their public suffix. There are use-cases that require operators of public suffix domains to demonstrate control over their domain, such as to be added to the Public Suffix List (Appendix A.1.1.4) or to provision a wildcard certificate. At the same time, if an operator of such a domain allows its customers or tenants to create names starting with an underscore ("_") then it opens up substantial risk to the domain operator for attackers to provision services on their domain.

Whether or not it is appropriate to allow domain verification on a public suffix will depend on the application. In the general case:

* Providers SHOULD NOT allow verification of ownership for domains which are public suffixes in the "ICANN" division. For example, "_foo-challenge.co.uk" would not be allowed.

* Providers MAY allow verification of ownership for domains which are public suffixes in the "PRIVATE" division, although it would be preferable to apply additional safety checks in this case.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[RFC1464] Rosenbaum, R., "Using the Domain Name System To Store Arbitrary String Attributes", RFC 1464, DOI 10.17487/RFC1464, May 1993, <https://www.rfc-editor.org/rfc/rfc1464>.

[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>.

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

[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>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/rfc/rfc9364>.

8.2. Informative References

[ACM-CNAME] AWS, "Option 1: DNS Validation", n.d., <https://docs.aws.amazon.com/acm/latest/userguide/dns-validation.html>.

[ACME-SCOPED-CHALLENGE] Chariton, A. A., Omidi, A. A., Kasten, J., Loukos, F., and S. A. Janikowski, "ACME Scoped DNS

Challenges", 2024, <https://datatracker.ietf.org/doc/draft-ietf-acme-scoped-dns-challenges/>.

[AKAMAI-DELEGATED] Akamai Technologies, "Onboard a secure by default property", 2023, <https://techdocs.akamai.com/property-mgr/reference/onboard-a-secure-by-default-property>.

[ATLASSIAN-VERIFY] Atlassian, "Verify over DNS", n.d., <https://support.atlassian.com/user-management/docs/verify-a-domain-to-manage-accounts/#Verify-over-DNS>.

[AVOID-FRAGMENTATION] Fujiwara, K. and P. Vixie, "Fragmentation Avoidance in DNS", 2023, <https://datatracker.ietf.org/doc/draft-ietf-dnsop-avoid-fragmentation/>.

[CLOUDFLARE-DELEGATED] Cloudflare, "Auto-renew TLS certificates with DCV Delegation", 2023, <https://blog.cloudflare.com/introducing-dcv-delegation/>.

[DNS-01] Let's Encrypt, "Challenge Types: DNS-01 challenge", 2020, <https://letsencrypt.org/docs/challenge-types/#dns-01-challenge>.

[DOCUSIGN-CNAME] DocuSign Admin for Organization Management, "Claim a Domain", n.d., <https://support.docusign.com/s/document-item?rsc_301=&bundleId=rrf1583359212854&topicId=gso1583359141256_1.html>.

[GITHUB-TXT] GitHub, "Verifying your organization's domain", n.d., <https://docs.github.com/en/github/setting-up-and-managing-organizations-and-teams/verifying-your-organizations-domain>.

[GOOGLE-WORKSPACE-CNAME] Google, "CNAME record values", n.d., <https://support.google.com/a/answer/112038>.

[GOOGLE-WORKSPACE-TXT] Google, "TXT record values", n.d., <https://support.google.com/a/answer/2716802>.

[I-D.draft-tjw-dbound2-problem-statement] Wicinski, T., "Domain Boundaries 2.0 Problem Statement", Work in Progress, Internet-Draft, draft-tjw-dbound2-problem-statement-01, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-tjw-dbound2-problem-statement-01>.

[LETSENCRYPT-90-DAYS-RENEWAL] Let's Encrypt, "Why ninety-day lifetimes for certificates?", 2015, <https://letsencrypt.org/2015/11/09/why-90-days.html>.

[PSL] Mozilla Foundation, "Public Suffix List", 2022, <https://publicsuffix.org/>.

[PSL-DIVISIONS] Frakes, J., "Public Suffix List format", 2022, <https://github.com/publicsuffix/list/wiki/Format#divisions>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/rfc/rfc3339>.

[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/rfc/rfc4086>.

[RFC4343] Eastlake 3rd, D., "Domain Name System (DNS) Case Insensitivity Clarification", RFC 4343, DOI 10.17487/RFC4343, January 2006, <https://www.rfc-editor.org/rfc/rfc4343>.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <https://www.rfc-editor.org/rfc/rfc6672>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/rfc/rfc8499>.

[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>.

[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>.

[RFC9210] Kristoff, J. and D. Wessels, "DNS Transport over TCP - Operational Requirements", BCP 235, RFC 9210, DOI 10.17487/RFC9210, March 2022, <https://www.rfc-editor.org/rfc/rfc9210>.

[SUBDOMAIN-TAKEOVER] Mozilla, "Subdomain takeovers", n.d., <https://developer.mozilla.org/en-US/docs/Web/Security/Subdomain_takeovers>.

Appendix A. Appendix

A survey of several different methods deployed today for DNS based domain control validation follows.

A.1. Survey of Techniques

A.1.1. TXT based

TXT records is usually the default option for domain control validation. The service provider asks the user to add a DNS TXT record (perhaps through their domain host or DNS provider) at the domain with a certain value. Then the service provider does a DNS TXT query for the domain being verified and checks that the correct value is present. For example, this is what a DNS TXT record could look like for a provider Foo:

```
example.com.   IN   TXT   "237943648324687364"
```

Here, the value "237943648324687364" serves as the randomly-generated TXT value being added to prove ownership of the domain to Foo provider. Note that in this construction provider Foo would have to query for all TXT records at "example.com" to get the validating record. Although the original DNS protocol specifications did not associate any semantics with the DNS TXT record, [RFC1464] describes how to use them to store attributes in the form of ASCII text key-value pairs for a particular domain. In practice, there is wide variation in the content of DNS TXT records used for domain control validation, and they often do not follow the key-value pair model. Even so, the RDATA [RFC1034] portion of the DNS TXT record has to contain the value being used to verify the domain. The value is usually a Random Token in order to guarantee that the entity who requested that the domain be verified (i.e. the person managing the account at Foo provider) is the one who has (direct or delegated) access to DNS records for the domain. After a TXT record has been added, the service provider will usually take some time to verify that the DNS TXT record with the expected token exists for the domain. The generated token typically expires in a few days.

Some providers use a prefix of _PROVIDER_NAME-challenge in the Name field of the TXT record challenge. For ACME, the full Host is _acme-challenge.<YOUR_DOMAIN>. Such patterns are useful for doing targeted domain control validation. The ACME protocol ([RFC8555]) has a challenge type DNS-01 that lets a user prove domain ownership. In this challenge, an implementing CA asks you to create a TXT record with a randomly-generated token at _acme-challenge.<YOUR_DOMAIN>:

```
_acme-challenge.example.com.  IN  TXT "cE3A8qQpEzAIYq-T9DWNdLJ1_YRXamdxc
```

[RFC8555] (section 8.4) places requirements on the Random Token.

A.1.1.1. Let's Encrypt

The ACME example in Appendix A.1.1 is implemented by Let's Encrypt [DNS-01].

A.1.1.2. Google Workspace

[GOOGLE-WORKSPACE-TXT] asks the user to sign in with their administrative account and obtain their token as part of the setup process for Google Workspace. The verification token is a 68-character string that begins with "google-site-verification=", followed by 43 characters. Google recommends a TTL of 3600 seconds. The owner name of the TXT record is the domain or subdomain name being verified.

A.1.1.3. GitHub

GitHub asks you to create a DNS TXT record under _github-challenge-ORGANIZATION.<YOUR_DOMAIN>, where ORGANIZATION stands for the GitHub organization name [GITHUB-TXT]. The code is a numeric code that expires in 7 days.

A.1.1.4. Public Suffix List

The Public Suffix List ([PSL]) asks for owners of private domains to authenticate by creating a TXT record containing the pull request URL for adding the domain to the Public Suffix List. For example, to authenticate "example.com" submitted under pull request 100, a requestor would add:

```
_psl.example.com.  IN TXT "https://github.com/publicsuffix/list/pull/100
```

A.1.2. CNAME based

A.1.2.1. CNAME for Domain Control Validation

A.1.2.1.1. DocuSign

[DOCUSIGN-CNAME] asks the user to add a CNAME record with the "Host Name" set to be a 32-digit random value pointing to verifydomain.docusign.net..

A.1.2.1.2. Google Workspace

[GOOGLE-WORKSPACE-CNAME] lets you specify a CNAME record for verifying domain ownership. The user gets a unique 12-character string that is added as "Host", with TTL 3600 (or default) and Destination an 86-character string beginning with "gv-" and ending with ".domainverify.googlehosted.com.".

A.1.2.2. Delegated Domain Control Validation

A.1.2.2.1. Content Delivery Networks (CDNs): Akamai and Cloudflare

In order to be issued a TLS cert from a Certification Authority like Let’s Encrypt, the requester needs to prove that they control the domain. Typically, this is done via the [DNS-01] challenge. Let’s Encrypt only issues certs with a 90 day validity period for security reasons [LETSENCRYPT-90-DAYS-RENEWAL]. This means that after 90 days, the DNS-01 challenge has to be re-done and the random token has to be replaced with a new one. Doing this manually is error-prone. Content Delivery Networks like Akamai and Cloudflare offer to automate this process using a CNAME record in the user's DNS that points to the validation record in the CDN's zone ([AKAMAI-DELEGATED] and [CLOUDFLARE-DELEGATED]).

A.1.2.2.2. AWS Certificate Manager (ACM)

AWS Certificate Manager [ACM-CNAME] allows delegated domain control validation Section 5.3.2. The record name for the CNAME looks like:

```
`_<random-token1>.example.com.   IN   CNAME _<random-token2>.acm-valida
```

The CNAME points to:

```
`_<random-token2>.acm-validations.aws.   IN   TXT <random-token3>`
```

Here, the random tokens are used for the following:

- *<random-token1>: Unique sub-domain, so there's no clashes when looking up the validation record.
- *<random-token2>: Proves to ACM that the requester controls the DNS for the requested domain.
- *<random-token3>: The actual token being verified.

Note that if there are more than 5 CNAMEs being chained, then this method does not work.

A.1.2.3. Atlassian

Some services ask the DNS record to exist in perpetuity [ATLASSIAN-VERIFY]. If the record is removed, the user gets a limited amount of time to re-add it before they lose domain validation status.

A.1.2.4. Constraints on Domains and Subdomains

A.1.2.4.1. CAA records

While the ACME protocol ([RFC8555]) specifies a way to demonstrate ownership over a given domain, Certification Authorities are required to use it in-conjunction with [RFC8659] that specifies CAA records. CAA allows a domain owner to apply policy across a domain and its subdomains to limit which Certification Authorities may issue certificates.

Authors' Addresses

Shivan Sahib
Brave Software

Email: shivankaulsahib@gmail.com

Shumon Huque
Salesforce

Email: shuque@gmail.com

Paul Wouters
Aiven

Email: paul.wouters@aiven.io

Erik Nygren
Akamai Technologies

Email: erik+ietf@nygren.org