

Workgroup: Network Working Group

Internet-Draft:

draft-sahib-domain-verification-techniques-03

Published: 7 March 2022

Intended Status: Informational

Expires: 8 September 2022

Authors: S. Sahib                      S. Huque              P. Wouters

          Brave Software      Salesforce      Aiven

## **Survey of Domain Verification Techniques using DNS**

### **Abstract**

Many services on the Internet need to verify ownership or control of a domain in the Domain Name System (DNS) [RFC1034] [RFC1035]. This verification is often done by requesting a specific DNS record to be visible in the domain. This document surveys various techniques in wide use today, the pros and cons of each, and proposes some practises 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/ShivanKaul/draft-sahib-domain-verification-techniques>.

### **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 8 September 2022.

### **Copyright Notice**

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Conventions and Definitions](#)
- [3. Verification Techniques](#)
  - [3.1. TXT based](#)
    - [3.1.1. Examples](#)
  - [3.2. CNAME based](#)
    - [3.2.1. Examples](#)
  - [3.3. Common Patterns](#)
    - [3.3.1. Name](#)
    - [3.3.2. RDATA](#)
- [4. Recommendations](#)
  - [4.1. Targeted Domain Verification](#)
  - [4.2. Targeted Service Verification](#)
  - [4.3. TXT vs CNAME](#)
  - [4.4. Time-bound checking](#)
- [5. Email sending authorization](#)
- [6. Security Considerations](#)
- [7. Operational Considerations](#)
- [8. IANA Considerations](#)
- [9. References](#)
  - [9.1. Normative References](#)
  - [9.2. Informative References](#)
- [Acknowledgments](#)
- [Authors' Addresses](#)

## 1. Introduction

Many providers of internet services need domain owners to prove that they control a particular domain before they can operate a services or grant some privilege to the associated domain. For instance, certificate authorities like Let's Encrypt [[LETSencrypt](#)] 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 domain control. This document describes and recommends common practises with using DNS based techniques for domain verification. Other techniques such as email or HTTP(S) based verification are out-of-scope.

In practice, DNS-based verification takes the form of the provider generating a random value visible only to the requester, and then asking the requester to create a DNS record containing this random value 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.

## 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.

Provider: an internet-based provider of a service, for e.g., Let's Encrypt provides a certificate authority service or GitHub provides code-hosting services. These services often require a user to verify that they control a domain.

## 3. Verification Techniques

### 3.1. TXT based

TXT record-based DNS domain verification is usually the default option for DNS verification. 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 value exists. For example, this is what a DNS TXT verification record could look like:

```
example.com.  IN  TXT  "foo-verification=bar-237943648324687364"
```

Here, the value "bar-237943648324687364" for the attribute "foo-verification" serves as the randomly-generated TXT value being added to prove ownership of the domain to Foo provider. 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 verification, and they often do not follow the key-value pair model. Even so, the rdata portion of the DNS TXT record has to contain the value being used to verify the domain. The value is usually a randomly-generated 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. The

generated token typically expires in a few days. The TXT record is placed at the domain being verified ("example.com" in the example above). 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 same domain name can have multiple distinct TXT records (a TXT Record Set), where each TXT record may be associated with a distinct service. Having many of these may cause operational issues, and it is RECOMMENDED that providers use a prefix (eg "\_foo.example.com") instead of using the top of the domain ("APEX") directly, such as:

```
_foo.example.com. IN TXT "bar-237943648324687364"
```

### 3.1.1. Examples

#### 3.1.1.1. Let's Encrypt

Let's Encrypt [[LETSENCRYPT](#)] has a challenge type DNS-01 that lets a user prove domain ownership in accordance with the ACME protocol [[RFC8555](#)]. In this challenge, Let's Encrypt asks you to create a TXT record with a randomly-generated token at \_acme-challenge.<YOUR\_DOMAIN>. For example, if you wanted to prove domain ownership of example.com, Let's Encrypt could ask you to create the DNS record:

```
_acme-challenge.example.com. IN TXT "cE3A8qQpEzAIYq-T9DWNdLJ1_YRXa
```

[[RFC8555](#)] (section 8.4) places requirements on the random value.

#### 3.1.1.2. Google Workspace

[[GOOGLE-WORKSPACE-TXT](#)] asks the user to sign in with their administrative account and obtain their verification 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.

#### 3.1.1.3. 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.

## 3.2. CNAME based

Less commonly than TXT record verification, service providers also provide the ability to verify domain ownership via CNAME records. One reason for using CNAME is for the case where the user cannot create TXT records. One common reason is that the domain name may already have CNAME record that aliases it to a 3rd-party target domain. CNAMEs have a technical restriction that no other record types can be placed along side them at the same domain name ([RFC1034], Section 3.6.2).. The CNAME based domain verification method typically uses a randomized label prepended to the domain name being verified.

### 3.2.1. Examples

#### 3.2.1.1. Google

[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."

To verify a subdomain, the unique 12-character string is appended with the subdomain name for "Host" field for e.g.

JLKDER712AFP.subdomain where subdomain is the subdomain being verified.

#### 3.2.1.2. AWS Certificate Manager (ACM)

To get issued a certificate by AWS Certificate Manager (ACM), you can create a CNAME record to verify domain ownership [ACM-CNAME]. The record name for the CNAME looks like:

```
`\_<random-token1>.example.com.  IN  CNAME  \_RANDOM-TOKEN.acm-validat
```

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

## 3.3. Common Patterns

### 3.3.1. Name

ACME and GitHub have a suffix of `_PROVIDER_NAME-challenge` in the Name field of the TXT record challenge. For ACME, the full Host is `_acme-challenge.<YOUR_DOMAIN>`, while for GitHub it is `_github-challenge-ORGANIZATION-<YOUR_DOMAIN>`. Both these patterns are useful for doing targeted domain verification, as discussed in section (#targeted-domain-verification) because if the provider knows what it is looking for (domain in the case of ACME, organization name + domain in case

of GitHub) it can specifically do a DNS query for that TXT record, as opposed to having to do a TXT query for the apex.

ACME does the same name construction for CNAME records.

### **3.3.2. RDATA**

One pattern that quite a few providers follow (Dropbox, Atlassian) is constructing the rdata of the TXT DNS record in the form of PROVIDER-SERVICE-domain-verification= followed by the random value being checked for. This is in accordance with [\[RFC1464\]](#) which mandates that attributes must be stored as key-value pairs.

## **4. Recommendations**

### **4.1. Targeted Domain Verification**

The TXT record being used for domain verification is most commonly placed at the domain name being verified. For example, if example.com is being verified, then the DNS TXT record will have example.com in the Name section. Unfortunately, this practise does not scale very well.

Many services are now attempting to verify domain names, causing many of these TXT records to be placed at that same location at the top of the domain (the APEX).

When a DNS administrator sees 15 DNS TXT records for their domain based on only random letters, they can no longer determine for which service or vendor the DNS TXT records were added. This causes administrators to leave all DNS TXT records in there, as they want to avoid breaking a service. Over time, the domain ends up with a lot of no longer needed, unknown and untracable DNS TXT records.

An operational issue arises from the DNS protocol only being able to query for "all TXT records" at a single location. If multiple services all require TXT records, this can cause the DNS answer for TXT records to become very large. It has been observed that some well known domains had so many services deployed that their DNS TXT answer did not fit in a single UDP DNS packet. This results in fragmentation which is known to be vulnerable to various attacks [draft-ietf-dnsop-avoid-fragmentation-06](#). It can also lead to UDP packet truncation, causing a retry over TCP. Not all networks properly transport DNS over TCP and some DNS software mistakenly believe TCP support is optional [draft-ietf-dnsop-dns-tcp-requirements-15](#).

### **4.2. Targeted Service Verification**

One malicious service that promises to deliver something after domain verification 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. The LetsEncrypt ACME challenge uses this method.

### 4.3. TXT vs CNAME

The inherent problem of a CNAME is that it cannot co-exist with any other data. What happens when both a CNAME and other data such as a TXT record or NS record exist depends on the DNS implementation. But most likely, either the CNAME or the other records will be silently ignored. The user interface for adding a record might not check for this. It might also break in unexpected ways. If a CNAME is added for continuous authorization, and for another service a TXT record is added, the TXT record might work but the CNAME record might break. Operational experience has also shown a vendor that provides two difference services, one requiring a CNAME and one requiring a TXT record for authorization that needed to be deployed at the same location. If both services would have used a TXT record, this would not have caused any problems.

Another issues with CNAME records is that they MUST NOT point to another CNAME. But where this might be true in an initial deployment, if the target that the CNAME points to is changed from a non-CNAME record to a CNAME record, some DNS software might no longer resolve this as expected.

Early web based DNS administration tools did not always have the TXT record available in a pulldown menu for DNS record types, while CNAME would be available. However as many anti-spam measures now require TXT records, this support is now generally available. It is recommended that the CNAME method is only used for delegating authorization to an actual subdomain, for example:

```
recrutement.example.com.    IN    CNAME    example.recrutement-vendor.co
```

### 4.4. Time-bound checking

After domain verification is done, there is no need for the TXT or CNAME record to continue to exist as the presence of the domain-verifying 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 verifying DNS record once the verification is done and the service provider doing the verification should specify how long the verification will take (i.e. after how much time can the verifying DNS record be deleted). However, despite this, some services ask the record to exist in perpetuity [[ATLASSIAN-VERIFY](#)].

If a provider will use the DNS TXT record only for a one-time verification, it is RECOMMENDED that they clearly indicate this in the RDATA of the TXT record, so a DNS administrator at the target domain can easily spot an obsolete record in the future. For example:

```
_provider-token.example.com. IN TXT "type=activation_only  
expiry=2023-10-12 token=TOKENDATA"
```

If a provider requires the continued presence of the TXT record as proof that the domain owner is still authorizing the service, this should also be clear from the TXT record RDATA. For example:

```
_provider-service.example.com. IN TXT "type=continued_service  
expiry=never token=TOKENDATA"
```

## 5. Email sending authorization

Some vendors use a hosted service that wants to generate emails that appear to be from the customer. When a customer has deployed anti-spam measures such as DKIM [[RFC6376](#)], DMARC [[RFC7489](#)] or SPF [[RFC7208](#)], the vendor's mail service needs to be added to the list of allowed mail servers. However, some customers might not want to give permission for a vendor to send emails from their entire domain. It is recommended that a vendor uses a subdomain. If the vendor's domain is example-vendor.com, and the customer domain is example-customer.com, the vendor could use the subdomain example-customer.example-vendor.com to send emails. Alternatively, the customer could delegate a subdomain example-vendor.example-customer.com to the vendor for email sending, as those email addresses would have a stronger origin appearance of being emails sent by the customer to their clients.

Besides requiring proof of ownership of the domain, the customer needs to authorize the hosted service to send email on their behalf.

## 6. Security Considerations

Both the provider and the service being authenticated and authorized should be obvious from the TXT content to prevent malicious services from misleading the domain owner into certifying a different provider or service.



It is RECOMMENDED that DNSSEC [[RFC4033](#)] is employed by the domain owner. A service provider MUST enable DNSSEC validation when verifying domain name challenges to protect against domain name spoofing.

## 7. Operational Considerations

It is often consumers of the provider services that are not DNS experts that need to relay information from a provider's website to their local DNS administrators. The exact DNS record type, content and location is often not clear when the DNS administrator receives the information. It is RECOMMENDED that providers offer extremely 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. It is recommended that any instructions given by the provider contains the entire DNS record using a Fully Qualified Domain Name (FQDN).

## 8. IANA Considerations

This document has no IANA actions.

## 9. References

### 9.1. 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>>.
- [[RFC1035](#)] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/rfc/rfc1035>>.
- [[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>>.
- [[RFC4033](#)] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC

4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/rfc/rfc4033>>.

[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>>.

## 9.2. Informative References

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

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

[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>>.

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

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <<https://www.rfc-editor.org/rfc/rfc6376>>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <<https://www.rfc-editor.org/rfc/rfc7208>>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/rfc/rfc7489>>.

[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>>.

## **Acknowledgments**

TODO

## **Authors' Addresses**

Shivan Sahib  
Brave Software

Email: [shivankaulsahib@gmail.com](mailto:shivankaulsahib@gmail.com)

Shumon Huque  
Salesforce

Email: [shuque@gmail.com](mailto:shuque@gmail.com)

Paul Wouters  
Aiven

Email: [paul.wouters@aiven.io](mailto:paul.wouters@aiven.io)