Workgroup: Network Working Group
Internet-Draft:
draft-sahib-domain-verification-techniques-02
Published: 10 June 2021
Intended Status: Informational
Expires: 12 December 2021
Authors: S. Sahib
Brave Software
S. Huque
Salesforce

Survey of Domain Verification Techniques using DNS

Abstract

Many services on the Internet need to verify ownership or control of domains in the Domain Name System (DNS) [RFC1034] [RFC1035]. This verification often relies on adding or editing DNS records within the domain. This document surveys various techniques in wide use today, the pros and cons of each, and possible improvements.

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 12 December 2021.

(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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Many providers on the internet need users to prove that they control a particular domain before granting them some sort of privilege associated with that domain. For instance, certificate authorities like Let's Encrypt [LETSENCRYPT] ask requesters of TLS certificates to prove that they operate the domain they're requesting the certificate for. Providers generally allow for several different ways of proving domain control, some of which include manipulating DNS records. This document focuses on DNS techniques for domain verification; other techniques (such as email or HTML verification) are out-of-scope.

In practice, DNS-based verification often 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 that the provider can

query for. Generally only one temporary DNS record is sufficient for proving domain ownership.

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-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 usually 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.

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 neme 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. This is used in case 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, which would point to _<random-token2>.<random-token3>.acm-validations.aws.

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.

If many services are attempting to verify the domain name, many distinct TXT records end up being placed at that name. There is no way to surgically query only the TXT record for a specific service, resulting in extra work for a verifying service to sift through the records for its own domain verification record. In addition, since DNS Resource Record sets are treated atomically, all TXT records must be returned to the querier, which leads to a bloating of DNS responses. This could cause truncation and retrying DNS queries over TCP, which is more resource intensive.

A better method is to place the TXT record at a subdomain of the domain being verified that is specially reserved for use by the application service in question. The LetsEncrypt ACME challenge mentioned earlier uses this method.

4.2. TXT vs CNAME

TODO

4.3. 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].

5. Security Considerations

DNSSEC [RFC4033] should be employed by the domain owner to protect against domain name spoofing.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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>.

7.2. 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>.

[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

Shumon Huque
Salesforce

Email: shuque@gmail.com