Workgroup: TLS Working Group
Internet-Draft: draft-nygren-tls-ip-in-sni-00
Published: 27 July 2022
Intended Status: Standards Track
Expires: 28 January 2023
Authors: E. Nygren R. Salz
Akamai Technologies Akamai Technologies

Representing IP addresses in TLS Server Name Indication (SNI)

Abstract

This specification provides a mechanism for clients to send IP addresses in a TLS Server Name Indication (SNI) extension as part of TLS handshakes, allowing servers to return a certificate containing that subjectAltName. This is done by converting the IP address to a special-use domain name.

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 28 January 2023.

Table of Contents

1. Introduction

TLS [RFC8446] clients often need to send a Server Name Indication (SNI) extension [RFC6066], Section 3 as part of their ClientHello message. This helps servers select a certificate to return that includes a subjectAltName which includes the SNI value. Without SNI, multi-tenant services need need as many IP addresses as server certificates, which is not generally a problem with IPv6, but is complicated by, as well as contributes to, address scarcity in IPv4.

Certificate subjectAltName (SAN) [RFC5280], Section 4.2.1.6 values can encode IP addresses (with a defined form for "iPAddress" that encodes both IPv4 and IPv6 addresses as a sequence of octets). However, the ServerName structure for SNI only defines "host_name" as a "name_type" and encoding the hostname in Hostname, and it does not specify a way to encode IP addresses.

The lack of support for IP addresses in SNI values is less problematic in the case where a client is connecting to the IP address that it expects to see in the certificate. However, some specifications such as [I-D.draft-ietf-add-ddr] have clients require that a particular IP address is present in the SNI while connecting to a different IP address.

This specification does NOT change any behaviors for how clients to validate certificates.

2. Notational Conventions

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

3. Indicating IP Addresses in SNI

TLS clients connecting to a server and expecting to find a given IP address in a certificate subjectAltName MUST encode the iPAddress they expect to find in the certificate into the SNI HostName field. This encoding is done by converting the IP address into its reverse DNS address, as also done in [RFC8738].

*Note that this encoding only applies to how IP addresses are represented in SNI and does *NOT* change how IP addresses are represented in certificate SANs.*

Clients encode IPv6 "ip6.arpa" [RFC3596] names, using a reverse mapping of the IP address as the HostName field. For example, if the IP address being validated is 2001:db8::1, the SNI HostName field whould contain "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa".

Clients encode IPv4 addresses as "in-addr.arpa" [RFC1034] names, using a reverse mapping of the address octets. For example, if the IPv4 address being validated is 192.0.2.7, the SNI HostName field would contain "7.2.0.192.in-addr.arpa".

Servers receiving a SNI HostName field with one of these ".arpa" names implement this specification by returning a certificate with a subjectAltName containing the corresponding IP address as an iPAddress, when such a certificate is available. Servers MUST ignore malformed ip6.arpa and in-addr.arpa SNI values, such as those which do not contain 34 or 6 labels, respectively. In the corner-case where a server has both a certificate with an iPAddress SAN matching the supplied SNI as well as a dNSName SAN that matches the .arpa SNI string, the server SHOULD return the former (the cert corresponding to the iPAddress SAN).

Note that there is no way to represent IP address prefixes in certificates subjectAltNames.

4. Rejected Alternatives

(Note to editor: this section is to be removed moved to an appendix or removed prior to publication.)

Two other approaches have been considered but rejected.

4.1. Alternative: New NameType

One approach would be to introduce ip_address as a new name_type (or perhaps one for each of IPv6 and IPv4). For example, something like:

```
struct {
    NameType name_type;
    select (name_type) {
        case host_name: HostName;
        case ip_address: IPAddress;
    } name;
} ServerName;
enum {
    host_name(0), ip_address(1), (255)
} NameType;
opaque HostName<1..2^16-1>;
opaque IPAddress<1..2^8-1>;
```

While the cleanest approach, discussions with TLS library implementation maintainers indicate that this would be disruptive and have wide-reaching impact to long-stable APIs. It is likely that this extension point ossified long ago, and that middle-boxes and other software would also have problems here, as the SNI value is visible in-the-clear and some devices are known to inspect it.

4.2. Alternative: Shove an IP into Hostname

Serializing an IP address into a string and shoving it into the HostName value (eg, just putting in "192.0.2.7" or "2001:db8::1") might work, but those are not valid host names. Just because they can be serialized on the wire doesn't mean they won't result in unforseen breakage when abused in this manner.

5. IANA Considerations

No IANA registry changes are needed with this approach?

(TODO: Do we need to update anything to indicate this special use of in-addr.arpa and ip6.arpa?)

(The first other alternative might need a new registry if we decided to take that approach.)

6. Security Considerations

Overloading the in-addr.arpa and ip6.arpa names has potential for confusion if there are implementations that have odd behaviors here, or which try and use certificates with dNSName subjectAltNames containing those as hostnames.

As an example, some middleboxes (such as security appliances) may use the SNI value as a hostname to resolve and direct connections towards and this may have odd results when it is a .arpa address.

CAB Forum is considering updating their guidance to clarify that the issuance of certificates on those names is prohibited [cabforum.servercert.153].

General issues may exist with using IP addresses in certificate subjectAltNames, but a detailed analysis of this is outside the scope of this specification. Beyond not supporting IP addresses in SNI fields, there may be issues in other areas:

* The lifespan of IP addresses may be highly variable. While the ownership of some IP addresses (such as well-known DNS public resolvers) may be quite static, many service providers issue IP addresses with very short lifetimes. Clients may rotate their IPv6 privacy addresses [RFC8981] every few hours, as a very widespread example.

* There is no way to use CAA records [RFC8659] to constrain certificates on IP addresses. While it may be wortth considering supporting CAA records within the in-addr.arpa and ip6.arpa name spaces to allow network operators to constrain certificate issuance on IP addresses under their control, that is outside of the scope of this specification.

Note that certificate Name Constraints [RFC5280], Section 4.2.1.10 do support IP addresses, but it is unclear how widely this is implemented by client validators. Private certificate authorities may wish to consider using Name Constraints to only allow issuance of IP address certificates to organizational IP space.

7. Privacy Considerations

The SNI extension is sent in cleartext on the network, and thus visible to a passive observer. Using [I-D.draft-ietf-tls-esni] Encrypted Client Hello to protect the SNI may help.

Similar issues that exist with hostname based SNI values (with being able to perform tracking and correlation) may exist with IP addresses in SNI as well.

There may also be protocol-specific risks when desired IP addresses are sent in-the-clear as SNI.

Note that in many cases, observers will also be able to see the IP address as the destination endpoint of connections.

8. Acknowledgments

Thank you to Kyle Rose, Jon Reed, Ben Kaduk, and others who provided valuable input towards this draft.

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/info/rfc1034>.

[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>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <https://www.rfc-editor.org/info/rfc3596>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

9.2. Informative References

[cabforum.servercert.153] "Clarify validation requirements for .arpa #153", n.d., <https://github.com/cabforum/servercert/issues/153>.

[I-D.draft-ietf-add-ddr] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", Work in Progress, Internet-Draft, draft-ietf-add-ddr-08, 5 July 2022, <https://www.ietf.org/archive/id/draft-ietf-add-ddr-08.txt>.

[I-D.draft-ietf-tls-esni] Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-14, 13 February 2022, <https://www.ietf.org/archive/id/draft-ietf-tls-esni-14.txt>.

[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>.

[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/info/rfc8659>.

[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>.

[RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <https://www.rfc-editor.org/info/rfc8981>.

Authors' Addresses

Erik Nygren
Akamai Technologies

Email: erik+ietf@nygren.org
URI: http://erik.nygren.org/

Rich Salz
Akamai Technologies

Email: rsalz@akamai.com