Workgroup: Network Working Group
Internet-Draft: draft-ietf-uta-rfc6125bis-02
Obsoletes: 6125 (if approved)
Published: 8 September 2021
Intended Status: Standards Track
Expires: 12 March 2022
Authors: P. Saint-Andre (Mozilla)
J. Hodges (Google)
R. Salz (Akamai Technologies)

Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)

Abstract

Many application technologies enable secure communication between two entities by means of Transport Layer Security (TLS) with Internet Public Key Infrastructure Using X.509 (PKIX) certificates. This document specifies procedures for representing and verifying the identity of application services in such interactions.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Using TLS in Applications Working Group mailing list (uta@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/uta/.

Source for this draft and an issue tracker can be found at https://github.com/richsalz/draft-ietf-uta-rfc6125bis.

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 March 2022.

Copyright Notice

Table of Contents

1. Introduction

1.1. Motivation

The visible face of the Internet largely consists of services that employ a client-server architecture in which an interactive or automated client communicates with an application service in order to retrieve or upload information, communicate with other entities, or access a broader network of services. When a client communicates with an application service using Transport Layer Security [TLS] or Datagram Transport Layer Security [DTLS], it references some notion of the server's identity (e.g., "the website at example.com") while attempting to establish secure communication. Likewise, during TLS negotiation, the server presents its notion of the service's identity in the form of a public-key certificate that was issued by a certification authority (CA) in the context of the Internet Public Key Infrastructure using X.509 [PKIX]. Informally, we can think of these identities as the client's "reference identity" and the server's "presented identity" (these rough ideas are defined more precisely later in this document through the concept of particular identifiers). In general, a client needs to verify that the server's presented identity matches its reference identity so it can authenticate the communication.

Many application technologies adhere to the pattern just outlined. Such protocols have traditionally specified their own rules for representing and verifying application service identity. Unfortunately, this divergence of approaches has caused some confusion among certification authorities, application developers, and protocol designers.

Therefore, to codify secure procedures for the implementation and deployment of PKIX-based authentication, this document specifies recommended procedures for representing and verifying application service identity in certificates intended for use in application protocols employing TLS.

1.2. Audience

The primary audience for this document consists of application protocol designers, who can reference this document instead of defining their own rules for the representation and verification of application service identity. Secondarily, the audience consists of certification authorities, service providers, and client developers from technology communities that might reuse the recommendations in this document when defining certificate issuance policies, generating certificate signing requests, or writing software algorithms for identity matching.

1.3. Changes since RFC 6125

This document revises and obsoletes [VERIFY] based on the decade of experience and changes since it was first published. The major changes, in no particular order, include:

* All references have been updated to the current latest version.

* The TLS SNI extension is no longer new, it is commonplace.

* The only legal place for a certificate wildcard name is as the left-most component in a domain name.

* It is no longer allowed to use the commonName RDN, known as CN-ID, to represent the server identity; only the subjectAltNames extension is used.

* References to the X.500 directory, the survey of prior art, and the sample text in Appendix A have been removed.

1.4. How to Read This Document

This document is longer than the authors would have liked because it was necessary to carefully define terminology, explain the underlying concepts, define the scope, and specify recommended behavior for both certification authorities and application software implementations. The following sections are of special interest to various audiences:

* Protocol designers might want to first read the checklist in Section 3.

- *Certification authorities might want to first read the recommendations for representation of server identity in Section 4.

- *Service providers might want to first read the recommendations for requesting of server certificates in Section 5.

- *Software implementers might want to first read the recommendations for verification of server identity in Section 6.

The sections on terminology (Section 1.8), naming of application services (Section 2), document scope (Section 1.7), and the like provide useful background information regarding the recommendations and guidelines that are contained in the above-referenced sections, but are not absolutely necessary for a first reading of this document.

1.5. Applicability

This document does not supersede the rules for certificate issuance or validation provided in [PKIX]. Therefore, [PKIX] is authoritative on any point that might also be discussed in this document. Furthermore, [PKIX] also governs any certificate-related topic on which this document is silent, including but not limited to certificate syntax, certificate extensions such as name constraints and extended key usage, and handling of certification paths.

This document addresses only name forms in the leaf "end entity" server certificate, not any name forms in the chain of certificates used to validate the server certificate. Therefore, in order to ensure proper authentication, application clients need to verify the entire certification path per [PKIX].

This document also does not supersede the rules for verifying service identity provided in specifications for existing application protocols published prior to this document. However, the procedures described here can be referenced by future specifications, including updates to specifications for existing application protocols if the relevant technology communities agree to do so.

1.6. Overview of Recommendations

To orient the reader, this section provides an informational overview of the recommendations contained in this document.

The previous version of this specification, [VERIFY], surveyed the current practice from many IETF standards and tried to generalize best practices. This document takes the lessons learned in the past decade and codifies them as best practices.

For the primary audience of application protocol designers, this document provides recommended procedures for the representation and verification of application service identity within PKIX certificates used in the context of TLS.

For the secondary audiences, in essence this document encourages certification authorities, application service providers, and application client developers to coalesce on the following practices:

- Stop including and checking strings that look like domain names in the subject's Common Name.
- Check DNS domain names via the subjectAlternativeName extension designed for that purpose: dNSName.
- Move toward including and checking even more specific subjectAlternativeName extensions where appropriate for using the protocol (e.g., uniformResourceIdentifier and the otherName form SRVName).
- Constrain and simplify the validation of wildcard certificates (e.g., a certificate containing an identifier for *.example.com).

1.7. Scope

1.7.1. In Scope

This document applies only to service identities associated with fully qualified DNS domain names, only to TLS and DTLS, and only to PKIX-based systems. As a result, the scenarios described in the following section are out of scope for this specification (although they might be addressed by future specifications).

1.7.2. Out of Scope

The following topics are out of scope for this specification:

- Client or end-user identities.

  Certificates representing client or end-user identities (e.g., the rfc822Name identifier) can be used for mutual authentication between a client and server or between two clients, thus enabling stronger client-server security or end-to-end security. However, certification authorities, application developers, and service operators have less experience with client certificates than with server certificates, thus giving us fewer models from which to generalize and a less solid basis for defining best practices.

- Identifiers other than fully qualified DNS domain names.

For example, this specification does not discuss IP addresses or other attributes within a certificate beyond the subjectAltName extension. The focus of this document is on application service identities, not specific resources located at such services. Therefore this document discusses Uniform Resource Identifiers [URI] only as a way to communicate a DNS domain name (via the URI "host" component or its equivalent), not as a way to communicate other aspects of a service such as a specific resource (via the URI "path" component) or parameters (via the URI "query" component).

*Security protocols other than [TLS] or [DTLS].

Although other secure, lower-layer protocols exist and even employ PKIX certificates at times (e.g., IPsec [IPSEC]), their use cases can differ from those of TLS-based and DTLS-based application technologies. Furthermore, application technologies have less experience with IPsec than with TLS, thus making it more difficult to gather feedback on proposed best practices.

*Keys or certificates employed outside the context of PKIX-based systems.

Some deployed application technologies use a web of trust model based on or similar to OpenPGP [OPENPGP], or use self-signed certificates, or are deployed on networks that are not directly connected to the public Internet and therefore cannot depend on Certificate Revocation Lists (CRLs) or the Online Certificate Status Protocol [OCSP] to check CA-issued certificates. However, the method for binding a public key to an identifier in OpenPGP differs essentially from the method in X.509, the data in self-signed certificates has not been certified by a third party in any way, and checking of CA-issued certificates via CRLs or OCSP is critically important to maintaining the security of PKIX-based systems. Attempting to define best practices for such technologies would unduly complicate the rules defined in this specification.

*Certification authority policies, such as:

  -What types or "classes" of certificates to issue and whether to apply different policies for them.

  -Whether to issue certificates based on IP addresses (or some other form, such as relative domain names) in addition to fully qualified DNS domain names.

  -Which identifiers to include (e.g., whether to include SRV-IDs or URI-IDs as defined in the body of this specification).

- How to certify or validate fully qualified DNS domain names and application service types.

- How to certify or validate other kinds of information that might be included in a certificate (e.g., organization name).

* Resolution of DNS domain names.

  Although the process whereby a client resolves the DNS domain name of an application service can involve several steps (e.g., this is true of resolutions that depend on DNS SRV resource records, Naming Authority Pointer (NAPTR) DNS resource records [NAPTR], and related technologies such as [S-NAPTR]), for our purposes we care only about the fact that the client needs to verify the identity of the entity with which it communicates as a result of the resolution process. Thus the resolution process itself is out of scope for this specification.

* User interface issues.

  In general, such issues are properly the responsibility of client software developers and standards development organizations dedicated to particular application technologies (see, for example, [WSC-UI]).

1.8. Terminology

Because many concepts related to "identity" are often too vague to be actionable in application protocols, we define a set of more concrete terms for use in this specification.

application service: A service on the Internet that enables interactive and automated clients to connect for the purpose of retrieving or uploading information, communicating with other entities, or connecting to a broader network of services.

application service provider: An organization or individual that hosts or deploys an application service.

application service type: A formal identifier for the application protocol used to provide a particular kind of application service at a domain; the application service type typically takes the form of a Uniform Resource Identifier scheme [URI] or a DNS SRV Service [DNS-SRV].

automated client: A software agent or device that is not directly controlled by a human user.

delegated domain: A domain name or host name that is explicitly configured for communicating with the source domain, by either

(a) the human user controlling an interactive client or (b) a trusted administrator. In case (a), one example of delegation is an account setup that specifies the domain name of a particular host to be used for retrieving information or connecting to a network, which might be different from the server portion of the user's account name (e.g., a server at mailhost.example.com for connecting to an IMAP server hosting an email address of juliet@example.com). In case (b), one example of delegation is an admin-configured host-to-address/address-to-host lookup table.

derived domain: A domain name or host name that a client has derived from the source domain in an automated fashion (e.g., by means of a [DNS-SRV] lookup).

identifier: A particular instance of an identifier type that is either presented by a server in a certificate or referenced by a client for matching purposes.

identifier type: A formally defined category of identifier that can be included in a certificate and therefore that can also be used for matching purposes. For conciseness and convenience, we define the following identifier types of interest, which are based on those found in the PKIX specification [PKIX] and various PKIX extensions.

*DNS-ID = a subjectAltName entry of type dNSName; see [PKIX]

*SRV-ID = a subjectAltName entry of type otherName whose name form is SRVName; see [SRVNAME]

*URI-ID = a subjectAltName entry of type uniformResourceIdentifier whose value includes both (i) a "scheme" and (ii) a "host" component (or its equivalent) that matches the "reg-name" rule (where the quoted terms represent the associated [ABNF] productions from [URI]); see [PKIX] and [URI]

interactive client: A software agent or device that is directly controlled by a human user. (Other specifications related to security and application protocols, such as [WSC-UI], often refer to this entity as a "user agent".)

pinning: The act of establishing a cached name association between the application service's certificate and one of the client's reference identifiers, despite the fact that none of the presented identifiers matches the given reference identifier. Pinning is accomplished by allowing a human user to positively accept the mismatch during an attempt to communicate with the application service. Once a cached name association is established, the certificate is said to be pinned to the

reference identifier and in future communication attempts the client simply verifies that the service's presented certificate matches the pinned certificate, as described under Section 6.6.2. (A similar definition of "pinning" is provided in [WSC-UI].)

PKIX: PKIX is a short name for the Internet Public Key Infrastructure using X.509 defined in RFC 5280 [PKIX], which comprises a profile of the X.509v3 certificate specifications and X.509v2 certificate revocation list (CRL) specifications for use in the Internet.

PKIX-based system: A software implementation or deployed service that makes use of X.509v3 certificates and X.509v2 certificate revocation lists (CRLs).

PKIX certificate: An X.509v3 certificate generated and employed in the context of PKIX.

presented identifier: An identifier that is presented by a server to a client within a PKIX certificate when the client attempts to establish secure communication with the server; the certificate can include one or more presented identifiers of different types, and if the server hosts more than one domain then the certificate might present distinct identifiers for each domain.

reference identifier: An identifier, constructed from a source domain and optionally an application service type, used by the client for matching purposes when examining presented identifiers.

Relative Distinguished Name (RDN): The ASN.1-based construction comprising a Relative Distinguished Name (RDN), which itself is a building-block component of Distinguished Names. See [LDAP-DN], Section 2.

source domain: The fully qualified DNS domain name that a client expects an application service to present in the certificate (e.g., www.example.com), typically input by a human user, configured into a client, or provided by reference such as in a hyperlink. The combination of a source domain and, optionally, an application service type enables a client to construct one or more reference identifiers.

subjectAltName entry: An identifier placed in a subjectAltName extension.

subjectAltName extension: A standard PKIX certificate extension [PKIX] enabling identifiers of various types to be bound to the certificate subject.

subject name:
In this specification, the term refers to the name of a PKIX certificate's subject, encoded in a certificate's subject field (see [PKIX], Section 4.1.2.6).

TLS client: An entity that assumes the role of a client in a Transport Layer Security [TLS] negotiation. In this specification we generally assume that the TLS client is an (interactive or automated) application client; however, in application protocols that enable server-to-server communication, the TLS client could be a peer application service.

TLS server: An entity that assumes the role of a server in a Transport Layer Security [TLS] negotiation; in this specification we assume that the TLS server is an application service.

Most security-related terms in this document are to be understood in the sense defined in [SECTERMS]; such terms include, but are not limited to, "attack", "authentication", "authorization", "certification authority", "certification path", "certificate", "credential", "identity", "self-signed certificate", "trust", "trust anchor", "trust chain", "validate", and "verify".

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.

2. Naming of Application Services

This section discusses naming of application services on the Internet, followed by a brief tutorial about subject naming in PKIX.

2.1. Naming Application Services

This specification assumes that the name of an application service is based on a DNS domain name (e.g., example.com) -- supplemented in some circumstances by an application service type (e.g., "the IMAP server at example.com").

From the perspective of the application client or user, some names are direct because they are provided directly by a human user (e.g., via runtime input, prior configuration, or explicit acceptance of a client communication attempt), whereas other names are indirect because they are automatically resolved by the client based on user input (e.g., a target name resolved from a source name using DNS SRV or NAPTR records). This dimension matters most for certificate consumption, specifically verification as discussed in this document.

From the perspective of the application service, some names are unrestricted because they can be used in any type of service (e.g., a certificate might be reused for both the HTTP service and the IMAP service at example.com), whereas other names are restricted because they can be used in only one type of service (e.g., a special-purpose certificate that can be used only for an IMAP service). This dimension matters most for certificate issuance.

Therefore, we can categorize the identifier types of interest as follows:

*A DNS-ID is direct and unrestricted.

*An SRV-ID is typically indirect but can be direct, and is restricted.

*A URI-ID is direct and restricted.

When implementing software, deploying services, and issuing certificates for secure PKIX-based authentication, it is important to keep these distinctions in mind. In particular, best practices differ somewhat for application server implementations, application client implementations, application service providers, and certification authorities. Ideally, protocol specifications that reference this document will specify which identifiers are mandatory-to-implement by servers and clients, which identifiers ought to be supported by certificate issuers, and which identifiers ought to be requested by application service providers. Because these requirements differ across applications, it is impossible to categorically stipulate universal rules (e.g., that all software implementations, service providers, and certification authorities for all application protocols need to use or support DNS-IDs as a baseline for the purpose of interoperability).

However, it is preferable that each application protocol will at least define a baseline that applies to the community of software developers, application service providers, and CAs actively using or supporting that technology (one such community, the CA/Browser Forum, has codified such a baseline for "Extended Validation Certificates" in [EV-CERTS]).

2.2. DNS Domain Names

For the purposes of this specification, the name of an application service is (or is based on) a DNS domain name that conforms to one of the following forms:

1. A "traditional domain name", i.e., a fully qualified DNS domain name or "FQDN" (see [DNS-CONCEPTS]) all of whose labels are "LDH labels" as described in [IDNA-DEFS]. Informally, such

labels are constrained to [US-ASCII] letters, digits, and the hyphen, with the hyphen prohibited in the first character position. Additional qualifications apply (please refer to the above-referenced specifications for details), but they are not relevant to this specification.

2. An "internationalized domain name", i.e., a DNS domain name that conforms to the overall form of a domain name (informally, dot-separated letter-digit-hyphen labels) but includes at least one label containing appropriately encoded Unicode code points outside the traditional US-ASCII range. That is, it contains at least one U-label or A-label, but otherwise may contain any mixture of NR-LDH labels, A-labels, or U-labels, as described in [IDNA-DEFS] and the associated documents.

2.3. Subject Naming in PKIX Certificates

For our purposes, an application service can be identified by a name or names carried in one or more of the following identifier types within subjectAltName entries:

*DNS-ID

*SRV-ID

*URI-ID

The Common Name RDN **MUST NOT** be used to identify a service. Reasons for this include:

*It is not strongly typed and therefore suffers from ambiguities in interpretation.

*It can appear more than once in the Subject Name.

For similar reasons, other RDN's within the Subject Name **MUST NOT** be used to identify a service.

3. Designing Application Protocols

This section provides guidelines for designers of application protocols, in the form of a checklist to follow when reusing the recommendations provided in this document.

*If your technology does not use DNS SRV records to resolve the DNS domain names of application services then consider stating that SRV-ID as defined in this document is not supported. Note that many existing application technologies use DNS SRV records to resolve the DNS domain names of application services, but do

not rely on representations of those records in PKIX certificates by means of SRV-IDs as defined in [SRVNAME].

* If your technology does not use use URIs to identify application services, then consider stating that URI-ID as defined in this document is not supported. Note that many existing application technologies use URIs to identify application services, but do not rely on representation of those URIs in PKIX certificates by means of URI-IDs.

* If your technology disallows the wildcard character in DNS domain names, then state the wildcard certificates as defined in this document are not supported.

4. Representing Server Identity

This section provides rules and guidelines for issuers of certificates.

4.1. Rules

When a certification authority issues a certificate based on the fully qualified DNS domain name at which the application service provider will provide the relevant application, the following rules apply to the representation of application service identities. The reader needs to be aware that some of these rules are cumulative and can interact in important ways that are illustrated later in this document.

1. The certificate **SHOULD** include a "DNS-ID" if possible as a baseline for interoperability.

2. If the service using the certificate deploys a technology for which the relevant specification stipulates that certificates ought to include identifiers of type SRV-ID (e.g., this is true of [XMPP]), then the certificate **SHOULD** include an SRV-ID.

3. If the service using the certificate deploys a technology for which the relevant specification stipulates that certificates ought to include identifiers of type URI-ID (e.g., this is true of [SIP] as specified by [SIP-CERTS], but not true of [HTTP] since [HTTP-TLS] does not describe usage of a URI-ID for HTTP services), then the certificate **SHOULD** include a URI-ID. The scheme **SHALL** be that of the protocol associated with the application service type and the "host" component (or its equivalent) **SHALL** be the fully qualified DNS domain name of the service. A specification that reuses this one **MUST** specify which URI schemes are to be considered acceptable in URI-IDs contained in PKIX certificates used for the application protocol (e.g., sip but not sips or tel for SIP as described in

[SIP-SIPS], or perhaps http and https for HTTP as might be described in a future specification).

4. The certificate **MAY** include other application-specific identifiers for types that were defined before publication of [SRVNAME] (e.g., XmppAddr for [XMPP]) or for which service names or URI schemes do not exist; however, such application-specific identifiers are not applicable to all application technologies and therefore are out of scope for this specification.

5. The certificate **MAY** contain more than one DNS-ID, SRV-ID, or URI-ID as further explained under Section 7.4.

4.2. Examples

Consider a simple website at www.example.com, which is not discoverable via DNS SRV lookups. Because HTTP does not specify the use of URIs in server certificates, a certificate for this service might include only a DNS-ID of www.example.com.

Consider an IMAP-accessible email server at the host mail.example.net servicing email addresses of the form user@example.net and discoverable via DNS SRV lookups on the application service name of example.net. A certificate for this service might include SRV-IDs of _imap.example.net and _imaps.example.net (see [EMAIL-SRV]) along with DNS-IDs of example.net and mail.example.net.

Consider a SIP-accessible voice-over-IP (VoIP) server at the host voice.example.edu servicing SIP addresses of the form user@voice.example.edu and identified by a URI of <sip:voice.example.edu>. A certificate for this service would include a URI-ID of sip:voice.example.edu (see [SIP-CERTS]) along with a DNS-ID of voice.example.edu.

Consider an XMPP-compatible instant messaging (IM) server at the host im.example.org servicing IM addresses of the form user@im.example.org and discoverable via DNS SRV lookups on the im.example.org domain. A certificate for this service might include SRV-IDs of _xmpp-client.im.example.org and _xmpp-server.im.example.org (see [XMPP]), a DNS-ID of im.example.org, and an XMPP-specific XmppAddr of im.example.org (see [XMPP]).

5. Requesting Server Certificates

This section provides rules and guidelines for service providers regarding the information to include in certificate signing requests (CSRs).

In general, service providers are encouraged to request certificates that include all of the identifier types that are required or recommended for the application service type that will be secured using the certificate to be issued.

If the certificate might be used for any type of application service, then the service provider is encouraged to request a certificate that includes only a DNS-ID.

If the certificate will be used for only a single type of application service, then the service provider is encouraged to request a certificate that includes a DNS-ID and, if appropriate for the application service type, an SRV-ID or URI-ID that limits the deployment scope of the certificate to only the defined application service type.

If a service provider offering multiple application service types (e.g., a World Wide Web service, an email service, and an instant messaging service) wishes to limit the applicability of certificates using SRV-IDs or URI-IDs, then the service provider is encouraged to request multiple certificates, i.e., one certificate per application service type. Conversely, the service provider is discouraged from requesting a single certificate containing multiple SRV-IDs or URI-IDs identifying each different application service type. This guideline does not apply to application service type "bundles" that are used to identify manifold distinct access methods to the same underlying application (e.g., an email application with access methods denoted by the application service types of imap, imaps, pop3, pop3s, and submission as described in [EMAIL-SRV]).

6. Verifying Service Identity

This section provides rules and guidelines for implementers of application client software regarding algorithms for verification of application service identity.

6.1. Overview

At a high level, the client verifies the application service's identity by performing the actions listed below (which are defined in the following subsections of this document):

1. The client constructs a list of acceptable reference identifiers based on the source domain and, optionally, the type of service to which the client is connecting.

2. The server provides its identifiers in the form of a PKIX certificate.

3. The client checks each of its reference identifiers against the presented identifiers for the purpose of finding a match.

4. When checking a reference identifier against a presented identifier, the client matches the source domain of the identifiers and, optionally, their application service type.

Naturally, in addition to checking identifiers, a client might complete further checks to ensure that the server is authorized to provide the requested service. However, such checking is not a matter of verifying the application service identity presented in a certificate, and therefore methods for doing so (e.g., consulting local policy information) are out of scope for this document.

6.2. Constructing a List of Reference Identifiers

6.2.1. Rules

The client **MUST** construct a list of acceptable reference identifiers, and **MUST** do so independently of the identifiers presented by the service.

The inputs used by the client to construct its list of reference identifiers might be a URI that a user has typed into an interface (e.g., an HTTPS URL for a website), configured account information (e.g., the domain name of a particular host or URI used for retrieving information or connecting to a network, which might be different from the DNS domain name portion of a username), a hyperlink in a web page that triggers a browser to retrieve a media object or script, or some other combination of information that can yield a source domain and an application service type.

The client might need to extract the source domain and application service type from the input(s) it has received. The extracted data **MUST** include only information that can be securely parsed out of the inputs (e.g., parsing the fully qualified DNS domain name out of the "host" component (or its equivalent) of a URI or deriving the application service type from the scheme of a URI) or information that is derived in a manner not subject to subversion by network attackers (e.g., pulling the data from a delegated domain that is explicitly established via client or system configuration, resolving the data via [DNSSEC], or obtaining the data from a third-party domain mapping service in which a human user has explicitly placed trust and with which the client communicates over a connection or association that provides both mutual authentication and integrity checking). These considerations apply only to extraction of the source domain from the inputs; naturally, if the inputs themselves are invalid or corrupt (e.g., a user has clicked a link provided by

a malicious entity in a phishing attack), then the client might end up communicating with an unexpected application service.

For example, given an input URI of <sips:alice@example.net>, a client would derive the application service type sip from the scheme and parse the domain name example.net from the host component.

Each reference identifier in the list **SHOULD** be based on the source domain and **SHOULD NOT** be based on a derived domain (e.g., a host name or domain name discovered through DNS resolution of the source domain). This rule is important because only a match between the user inputs and a presented identifier enables the client to be sure that the certificate can legitimately be used to secure the client's communication with the server. There is only one scenario in which it is acceptable for an interactive client to override the recommendation in this rule and therefore communicate with a domain name other than the source domain: because a human user has "pinned" the application service's certificate to the alternative domain name as further discussed under Section 6.6.4 and Section 7.1. In this case, the inputs used by the client to construct its list of reference identifiers might include more than one fully qualified DNS domain name, i.e., both (a) the source domain and (b) the alternative domain contained in the pinned certificate.

Using the combination of fully qualified DNS domain name(s) and application service type, the client constructs a list of reference identifiers in accordance with the following rules:

* The list **SHOULD** include a DNS-ID. A reference identifier of type DNS-ID can be directly constructed from a fully qualified DNS domain name that is (a) contained in or securely derived from the inputs (i.e., the source domain), or (b) explicitly associated with the source domain by means of user configuration (i.e., a derived domain).

* If a server for the application service type is typically discovered by means of DNS SRV records, then the list **SHOULD** include an SRV-ID.

* If a server for the application service type is typically associated with a URI for security purposes (i.e., a formal protocol document specifies the use of URIs in server certificates), then the list **SHOULD** include a URI-ID.

Which identifier types a client includes in its list of reference identifiers is a matter of local policy. For example, in certain deployment environments, a client that is built to connect only to a particular kind of service (e.g., only IM services) might be configured to accept as valid only certificates that include an SRV-

ID for that application service type; in this case, the client would include only SRV-IDs matching the application service type in its list of reference identifiers (not, for example, DNS-IDs). By contrast, a more lenient client (even one built to connect only to a particular kind of service) might include both SRV-IDs and DNS-IDs in its list of reference identifiers.

6.2.2. Examples

A web browser that is connecting via HTTPS to the website at www.example.com would have a single reference identifier: a DNS-ID of www.example.com.

A mail user agent that is connecting via IMAPS to the email service at example.net (resolved as mail.example.net) might have three reference identifiers: an SRV-ID of _imaps.example.net (see [EMAIL-SRV]), and DNS-IDs of example.net and mail.example.net. (A legacy email user agent would not support [EMAIL-SRV] and therefore would probably be explicitly configured to connect to mail.example.net, whereas an SRV-aware user agent would derive example.net from an email address of the form user@example.net but might also accept mail.example.net as the DNS domain name portion of reference identifiers for the service.)

A voice-over-IP (VoIP) user agent that is connecting via SIP to the voice service at voice.example.edu might have only one reference identifier: a URI-ID of sip:voice.example.edu (see [SIP-CERTS]).

An instant messaging (IM) client that is connecting via XMPP to the IM service at im.example.org might have three reference identifiers: an SRV-ID of _xmpp-client.im.example.org (see [XMPP]), a DNS-ID of im.example.org, and an XMPP-specific XmppAddr of im.example.org (see [XMPP]).

6.3. Preparing to Seek a Match

Once the client has constructed its list of reference identifiers and has received the server's presented identifiers in the form of a PKIX certificate, the client checks its reference identifiers against the presented identifiers for the purpose of finding a match. The search fails if the client exhausts its list of reference identifiers without finding a match. The search succeeds if any presented identifier matches one of the reference identifiers, at which point the client **SHOULD** stop the search.

Before applying the comparison rules provided in the following sections, the client might need to split the reference identifier

into its DNS domain name portion and its application service type portion, as follows:

* A reference identifier of type DNS-ID does not include an application service type portion and thus can be used directly as the DNS domain name for comparison purposes. As an example, a DNS-ID of www.example.com would result in a DNS domain name portion of www.example.com.

* For a reference identifier of type SRV-ID, the DNS domain name portion is the Name and the application service type portion is the Service. As an example, an SRV-ID of _imaps.example.net would be split into a DNS domain name portion of example.net and an application service type portion of imaps (mapping to an application protocol of IMAP as explained in [EMAIL-SRV]).

* For a reference identifier of type URI-ID, the DNS domain name portion is the "reg-name" part of the "host" component (or its equivalent) and the application service type portion is the application service type associated with the scheme name matching the [ABNF] "scheme" rule from [URI] (not including the ':' separator). As previously mentioned, this document specifies that a URI-ID always contains a "host" component (or its equivalent) containing a "reg-name". (Matching only the "reg-name" rule from [URI] limits verification to DNS domain names, thereby differentiating a URI-ID from a uniformResourceIdentifier entry that contains an IP address or a mere host name, or that does not contain a "host" component at all.) Furthermore, note that extraction of the "reg-name" might necessitate normalization of the URI (as explained in [URI]). As an example, a URI-ID of sip:voice.example.edu would be split into a DNS domain name portion of voice.example.edu and an application service type of sip (associated with an application protocol of SIP as explained in [SIP-CERTS]).

Detailed comparison rules for matching the DNS domain name portion and application service type portion of the reference identifier are provided in the following sections.

6.4. Matching the DNS Domain Name Portion

The client **MUST** match the DNS domain name portion of a reference identifier according to the following rules (and **SHOULD** also check the application service type as described under Section 6.5). The rules differ depending on whether the domain to be checked is a "traditional domain name" or an "internationalized domain name" (as defined under Section 2.2). Furthermore, to meet the needs of clients that support presented identifiers containing the wildcard

character "*", we define a supplemental rule for such "wildcard certificates".

6.4.1. Checking of Traditional Domain Names

If the DNS domain name portion of a reference identifier is a "traditional domain name", then matching of the reference identifier against the presented identifier is performed by comparing the set of domain name labels using a case-insensitive ASCII comparison, as clarified by [DNS-CASE] (e.g., WWW.Example.Com would be lower-cased to www.example.com for comparison purposes). Each label **MUST** match in order for the names to be considered to match, except as supplemented by the rule about checking of wildcard labels (Section 6.4.3).

6.4.2. Checking of Internationalized Domain Names

If the DNS domain name portion of a reference identifier is an internationalized domain name, then an implementation **MUST** convert any U-labels [IDNA-DEFS] in the domain name to A-labels before checking the domain name. In accordance with [IDNA-PROTO], A-labels **MUST** be compared as case-insensitive ASCII. Each label **MUST** match in order for the domain names to be considered to match, except as supplemented by the rule about checking of wildcard labels (Section 6.4.3; but see also Section 7.2 regarding wildcards in internationalized domain names).

6.4.3. Checking of Wildcard Certificates

A client **MAY** match the reference identifier against a presented identifier whose DNS domain name portion contains the wildcard character "*" in a label (following the description of labels and domain names in [DNS-CONCEPTS]), provided these requirements are met:

1. There is only one wildcard character.

2. The wildcard character appears only as the content of the left-most label.

3. The wildcard character is not embedded in an A-label or U-label [IDNA-DEFS] of an internationalized domain name [IDNA-PROTO].

A wildcard in a presented identifier can only match exactly one label in a reference identifier. Note that this is not the same as DNS wildcard matching, where the "*" label always matches at least one whole label and sometimes more. See [DNS-CONCEPTS], Section 4.3.3 and [DNS-WILDCARDS].

For information regarding the security characteristics of wildcard certificates, see Section 7.2.

6.5. Matching the Application Service Type Portion

When a client checks identifiers of type SRV-ID and URI-ID, it **MUST** check not only the DNS domain name portion of the identifier but also the application service type portion. The client does this by splitting the identifier into the DNS domain name portion and the application service type portion (as described under Section 6.3), then checking both the DNS domain name portion (as described under Section 6.4) and the application service type portion as described in the following subsections.

Implementation Note: An identifier of type SRV-ID or URI-ID provides an application service type portion to be checked, but that portion is combined only with the DNS domain name portion of the SRV-ID or URI-ID itself. For example, if a client's list of reference identifiers includes an SRV-ID of _xmpp-client.im.example.org and a DNS-ID of apps.example.net, the client would check (a) the combination of an application service type of xmpp-client and a DNS domain name of im.example.org and (b) a DNS domain name of apps.example.net. However, the client would not check (c) the combination of an application service type of xmpp-client and a DNS domain name of apps.example.net because it does not have an SRV-ID of _xmpp-client.apps.example.net in its list of reference identifiers.

6.5.1. SRV-ID

The application service name portion of an SRV-ID (e.g., imaps) **MUST** be matched in a case-insensitive manner, in accordance with [DNS-SRV]. Note that the _ character is prepended to the service identifier in DNS SRV records and in SRV-IDs (per [SRVNAME]), and thus does not need to be included in any comparison.

6.5.2. URI-ID

The scheme name portion of a URI-ID (e.g., sip) **MUST** be matched in a case-insensitive manner, in accordance with [URI]. Note that the : character is a separator between the scheme name and the rest of the URI, and thus does not need to be included in any comparison.

6.6. Outcome

The outcome of the matching procedure is one of the following cases.

6.6.1. Case #1: Match Found

If the client has found a presented identifier that matches a reference identifier, then the service identity check has succeeded. In this case, the client **MUST** use the matched reference identifier as the validated identity of the application service.

6.6.2. Case #2: No Match Found, Pinned Certificate

If the client does not find a presented identifier matching any of the reference identifiers but the client has previously pinned the application service's certificate to one of the reference identifiers in the list it constructed for this communication attempt (as "pinning" is explained under Section 1.8), and the presented certificate matches the pinned certificate (including the context as described under Section 7.1), then the service identity check has succeeded.

6.6.3. Case #3: No Match Found, No Pinned Certificate

If the client does not find a presented identifier matching any of the reference identifiers and the client has not previously pinned the certificate to one of the reference identifiers in the list it constructed for this communication attempt, then the client **MUST** proceed as described under Section 6.6.4.

6.6.4. Fallback

If the client is an interactive client that is directly controlled by a human user, then it **SHOULD** inform the user of the identity mismatch and automatically terminate the communication attempt with a bad certificate error; this behavior is preferable because it prevents users from inadvertently bypassing security protections in hostile situations.

Some interactive clients give advanced users the option of proceeding with acceptance despite the identity mismatch. Although this behavior can be appropriate in certain specialized circumstances, it needs to be handled with extreme caution, for example by first encouraging even an advanced user to terminate the communication attempt and, if the advanced user chooses to proceed anyway, by forcing the user to view the entire certification path before proceeding.

Otherwise, if the client is an automated application not directly controlled by a human user, then it **SHOULD** terminate the communication attempt with a bad certificate error and log the error appropriately. An automated application **MAY** provide a configuration setting that disables this behavior, but **MUST** enable the behavior by default.

7. Security Considerations

7.1. Pinned Certificates

As defined under Section 1.8, a certificate is said to be "pinned" to a DNS domain name when a user has explicitly chosen to associate a service's certificate with that DNS domain name despite the fact that the certificate contains some other DNS domain name (e.g., the user has explicitly approved apps.example.net as a domain associated with a source domain of example.com). The cached name association **MUST** take account of both the certificate presented and the context in which it was accepted or configured (where the "context" includes the chain of certificates from the presented certificate to the trust anchor, the source domain, the application service type, the service's derived domain and port number, and any other relevant information provided by the user or associated by the client).

7.2. Wildcard Certificates

Wildcard certificates, those that have an identifier with "*" as the left-most DNS label, automatically vouch for any single-label host names within their domain, but not multiple levels of domains. This can be convenient for administrators but also poses the risk of vouching for rogue or buggy hosts. See for example [Defeating-SSL] (beginning at slide 91) and [HTTPSbytes] (slides 38-40).

Protection against a wildcard that identifies a so-called "public suffix" (e.g., *.co.uk or *.com) is beyond the scope of this document.

7.3. Internationalized Domain Names

Allowing internationalized domain names can lead to the inclusion of visually similar (so-called "confusable") characters in certificates; for discussion, see for example [IDNA-DEFS].

7.4. Multiple Identifiers

A given application service might be addressed by multiple DNS domain names for a variety of reasons, and a given deployment might service multiple domains or protocols. The client **SHOULD** use the TLS Server Name Identification (SNI) extension as discussed in [TLS], Section 4.4.2.2.

To accommodate the workaround that was needed before the development of the SNI extension, this specification allows multiple DNS-IDs, SRV-IDs, or URI-IDs in a certificate.

8. References

8.1. Normative References

[DNS-CONCEPTS] Mockapetris, P.V., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://doi.org/10.17487/RFC1034>.

[DNS-SRV] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://doi.org/10.17487/RFC2782>.

[DNS-WILDCARDS] Lewis, E., "The Role of Wildcards in the Domain Name System", RFC 4592, DOI 10.17487/RFC4592, July 2006, <https://doi.org/10.17487/RFC4592>.

[IDNA-DEFS] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://doi.org/10.17487/RFC5890>.

[IDNA-PROTO] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, DOI 10.17487/RFC5891, August 2010, <https://doi.org/10.17487/RFC5891>.

[LDAP-DN] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, DOI 10.17487/RFC4514, June 2006, <https://doi.org/10.17487/RFC4514>.

[PKIX] 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://doi.org/10.17487/RFC5280>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://doi.org/10.17487/RFC2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://doi.org/10.17487/RFC8174>.

[SRVNAME] Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, DOI 10.17487/RFC4985, August 2007, <https://doi.org/10.17487/RFC4985>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://doi.org/10.17487/RFC3986>.

8.2. Informative References

[ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://doi.org/10.17487/RFC5234>.

[Defeating-SSL] Marlinspike, M., "New Tricks for Defeating SSL in Practice", BlackHat DC, February 2009, <http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf>.

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

[DNSSEC] 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://doi.org/10.17487/RFC4033>.

[DTLS] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://doi.org/10.17487/RFC6347>.

[EMAIL-SRV] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, DOI 10.17487/RFC6186, March 2011, <https://doi.org/10.17487/RFC6186>.

[EV-CERTS] CA/Browser Forum, "Guidelines For The Issuance And Management Of Extended Validation Certificates", October 2009, <http://www.cabforum.org/Guidelines_v1_2.pdf>.

[HTTP] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://doi.org/10.17487/RFC7230>.

[HTTP-TLS] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://doi.org/10.17487/RFC2818>.

[HTTPSbytes] Sokol, J. and R. Hansen, "HTTPS Can Byte Me", BlackHat Abu Dhabi, November 2010, <https://media.blackhat.com/bh-

ad-10/Hansen/Blackhat-AD-2010-Hansen-Sokol-HTTPS-Can-Byte-Me-slides.pdf>.

[IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://doi.org/10.17487/RFC4301>.

[NAPTR] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, DOI 10.17487/RFC3403, October 2002, <https://doi.org/10.17487/RFC3403>.

[OCSP] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://doi.org/10.17487/RFC6960>.

[OPENPGP] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <https://doi.org/10.17487/RFC4880>.

[S-NAPTR] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, DOI 10.17487/RFC3958, January 2005, <https://doi.org/10.17487/RFC3958>.

[SECTERMS] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://doi.org/10.17487/RFC4949>.

[SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://doi.org/10.17487/RFC3261>.

[SIP-CERTS] Gurbani, V., Lawrence, S., and A. Jeffrey, "Domain Certificates in the Session Initiation Protocol (SIP)", RFC 5922, DOI 10.17487/RFC5922, June 2010, <https://doi.org/10.17487/RFC5922>.

[SIP-SIPS] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, DOI 10.17487/RFC5630, October 2009, <https://doi.org/10.17487/RFC5630>.

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

[US-ASCII] American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

[VERIFY] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://doi.org/10.17487/RFC6125>.

[WSC-UI] Saldhana, A. and T. Roessler, "Web Security Context: User Interface Guidelines", World Wide Web Consortium LastCall WD-wsc-ui-20100309, March 2010, <http://www.w3.org/TR/2010/WD-wsc-ui-20100309>.

[XMPP] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <https://doi.org/10.17487/RFC6120>.

Acknowledgements

We gratefully acknowledge everyone who contributed to the previous version of this document, [VERIFY].

Authors' Addresses

Peter Saint-Andre
Mozilla
United States of America

Email: stpeter@mozilla.com

Jeff Hodges
Google
United States of America

Email: jdhodges@google.com

Rich Salz
Akamai Technologies
United States of America

Email: rsalz@akamai.com