Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 23, 2011

P. Saint-Andre
Cisco
J. Hodges
PayPal

October 20, 2010

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)

draft-saintandre-tls-server-id-check-10

Abstract

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

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 http://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 April 23, 2011.

Copyright Notice

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

1.1. Motivation

The visible face of the Internet consists of services that employ a client-server architecture in which an interactive or automated client connects to an application service in order to retrieve or upload information, communicate with other entities, or access a broader network of services. When a client connects to an application service using Transport Layer Security [TLS] or Datagram Transport Layer Security [DTLS], it references some conception of the server's identity while attempting to establish a secure connection (e.g., "the website at example.com"). Likewise, during TLS negotiation the server presents its conception 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 be sure that the certificate can legitimately be used to authenticate the connection.

Many application technologies adhere to the pattern just outlined, including but not limited to the following:

- The Internet Message Access Protocol [IMAP] and the Post Office

Protocol [POP3], for which see also [USINGTLS]

- The Hypertext Transfer Protocol [HTTP], for which see also [HTTP-TLS]
- The Lightweight Directory Access Protocol [LDAP], for which see also [LDAP-AUTH] and its predecessor [LDAP-TLS]
- The Simple Mail Transfer Protocol [SMTP], for which see also [SMTP-AUTH] and [SMTP-TLS]
- The Extensible Messaging and Presence Protocol [XMPP], for which see also [XMPP-OLD]
- The Network News Transfer Protocol [NNTP], for which see also [NNTP-TLS]

- The NETCONF Configuration Protocol [NETCONF], for which see also [NETCONF-SSH] and [NETCONF-TLS]
- The Syslog Protocol [SYSLOG], for which see also [SYSLOG-TLS] and [SYSLOG-DTLS]
- The Session Initiation Protocol [SIP], for which see also [SIP-CERTS]
- The Simple Network Management Protocol [SNMP], for which see also [SNMP-TLS]
- The General Internet Signalling Transport [GIST]

Application 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 best current practices regarding the implementation and deployment of secure PKIX-based authentication,

this document specifies recommended procedures for representing and verifying application service identity in certificates intended for use in applications employing TLS.

1.2. Applicability and Audience

This document does not supersede the rules for certificate issuance or validation provided in [PKIX], which governs any certificate-related topic on which this document is silent (e.g., certificate syntax, certificate extensions such as name constraints and extended key usage, and handling of certification paths). Specifically, in order to ensure proper authentication, application clients need to verify the entire certification path, because this document addresses only name forms in the leaf server certificate, not any name forms in the chain of certificates used to validate the server certificate.

This document also does not supersede the rules for verifying service identity provided in specifications for existing application protocols, such as those mentioned under Appendix A. However, the best current practices 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. It is also expected that this document will be updated or obsoleted in the future as best practices for issuance and verification of PKIX certificates continue to evolve through more widespread implementation and deployment of TLS-protected application

services over the Internet.

The primary audience for this document consists of application protocol designers, who might 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, client developers, service providers, and other members of technology communities that might re-use or "profile" the recommendations in this document when defining certificate issuance policies, writing software algorithms for identity matching, generating certificate signing requests, etc.

1.3. Overview of Recommendations

To orient the reader, this section provides an informational overview

of the recommendations contained in this document.

For the primary audience of application protocol designers, based on best current practices this document provides recommended procedures for 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 best current practices:

- Move away from including and checking strings that look like domain names in the subject's Common Name.

- Move toward including and checking 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 the using protocol (e.g., uniformResourceIdentifier and the otherName form SRVName).

- Move away from the issuance of so-called wildcard certificates (e.g., a certificate containing an identifier for "*.example.com").

These suggestions are not entirely consistent with all practices that are currently followed by certification authorities, client developers, and service providers. However, they reflect the best aspects of current practices and are expected to become more widely adopted in the coming years.

1.4. Scope

1.4.1. In Scope

This document applies only to service identities associated with fully-qualified DNS domain names, only to TLS and DTLS (or the older Secure Sockets Layer (SSL) technology), 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).

1.4.2. 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 gives us fewer models from which to generalize and a less solid basis for defining best practices.

- Identifiers other than fully-qualified DNS domain names.

  Some certification authorities issue server certificates based on IP addresses, but preliminary evidence indicates that such certificates are a very small percentage (less than 1%) of issued certificates. Furthermore, IP addresses are not necessarily reliable identifiers for application services because of the existence of private internets [PRIVATE], host mobility, multiple interfaces on a given host, Network Address Translators (NATs) resulting in different addresses for a host from different locations on the network, the practice of grouping many hosts together behind a single IP address, etc. Most fundamentally, most users find DNS domain names much easier to work with than IP addresses, which is why the domain name system was designed in the first place. We prefer to define best practices for the much more common use case and not to complicate the rules in this specification.

  Furthermore, we focus here on application service identities, not specific resources located at such services, e.g., a specific web page that can be accessed at a particular Uniform Resource Identifier [URI] whose authority component is the DNS domain name

of the application service. We also do not address identifiers

derived from Naming Authority Pointer (NAPTR) DNS resource records [NAPTR] and related technologies such as [S-NAPTR], since such identifiers cannot be validated in a trusted manner in the absence of [DNSSEC].

Finally, we do not discuss attributes unrelated to DNS domain names, such as those defined in [X.520] and other such specifications (e.g., organizational attributes, geographical attributes, company logos, and the like).

- Security protocols other than [TLS], [DTLS], or the older Secure Sockets Layer (SSL) technology.

  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 syntax of OpenPGP differs essentially from that of 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 OSCP 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.

Furthermore, this document also does not address various certification authority policies, such as:

- What types or "classes" of certificates to issue and whether to apply different policies for them (e.g., allow the wildcard character in certificates issued to individuals who have provided proof of identity but do not allow the wildcard character in "Extended Validation" certificates [EV-CERTS]).

- 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 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).

Finally, this specification is mostly silent about user interface issues, which in general are properly the responsibility of client software developers and standards development organizations dedicated to particular application technologies (see for example [WSC-UI]).

1.5. 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.

attribute-type-and-value pair: A colloquial name for the ASN.1-based construction comprising a Relative Distinguished Name (RDN), which itself is a building-block component of Distinguished Names. See Section 2 of [LDAP-DN].

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 connecting to 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). 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; the types covered in this specification are:

* CN-ID = a Relative Distinguished Name (RDN) in the certificate subject that contains one and only one attribute-type-and-value pair of type Common Name (CN); see [PKIX] and also [LDAP-SCHEMA]

* 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; see [PKIX]

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"; however that term is neither entirely accurate nor consistent with the terminology of common application protocols such as [HTTP].)

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 connect to the application service. Once a cached name association is established, the certificate is said to be pinned to the reference identifier and in future connection attempts the client simply verifies that the service's presented certificate matches the pinned certificate, as described under Section 4.6.2. (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 the server's PKIX certificate when the client attempts to establish a secure connection with the server; the certificate can include one or more presented identifiers of different types.

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

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

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, a 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 -- in addition to, or in place of, identifiers that may be embedded in or provided as a certificate's subject field.

subject field: The subject field of a PKIX certificate identifies the entity associated with the public key stored in the subject public key field (see Section 4.1.2.6 of [PKIX]).

subject name: In an overall sense, a subject's name(s) can be represented by or in the subject field, the subjectAltName extension, or both (see [PKIX] for details). More specifically, the term often refers to the composite name of a PKIX certificate's subject, encoded as the X.501 type Name and conveyed in a certificate's subject field (see Section 4.1.2.6 of [PKIX]).

TLS client: An entity that assumes the role of a client in a Transport Layer Security [TLS] negotiation; in this specfication 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 specfication 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 [KEYWORDS].

1.6. Contributors

The following individuals made important contributions to the text of this document: Shumon Huque, RL 'Bob' Morgan, and Kurt Zeilenga.

1.7. Acknowledgements

The editors and contributors wish to thank the following individuals for their feedback and suggestions: Bernard Aboba, Richard Barnes, Uri Blumenthal, Nelson Bolyard, Kaspar Brand, Anthony Bryan, Ben Campbell, Scott Cantor, Wan-Teh Chang, Bil Corry, Dave Cridland, Dave Crocker, Cyrus Daboo, Charles Gardiner, Philip Guenther, Phillip Hallam-Baker, Bruno Harbulot, Wes Hardaker, David Harrington, Paul Hoffman, Love Hornquist Astrand, Henry Hotz, Russ Housley, Jeffrey

Hutzelman, Simon Josefsson, Geoff Keating, John Klensin, Scott Lawrence, Matt McCutchen, Alexey Melnikov, Eddy Nigg, Ludwig Nussel, Joe Orton, Tom Petch, Yngve N. Pettersen, Tim Polk, Robert Relyea, Eric Rescorla, Pete Resnick, Martin Rex, Joe Salowey, Stefan Santesson, Jim Schaad, Rob Stradling, Michael Stroeder, Andrew Sullivan, Peter Sylvester, Martin Thomson, Paul Tiemann, Sean Turner, Nicolas Williams, Dan Wing, Dan Winship, and Stefan Winter.

2. Names

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 a 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 the user (e.g., via runtime input or prior configuration) whereas other names are indirect because they are resolved by the client based on input provided directly by the user (e.g., a target name resolved from a source name using DNS SRV records). This dimension matters most for certificate verification.

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 re-used 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:

- A CN-ID is direct (provided by a user) and unrestricted (can be used for any application).
- A DNS-ID is direct (provided by a user) and unrestricted (can be used for any application).
- An SRV-ID can be either direct (provided by a user) or more typically indirect (resolved by a client), and is restricted (can be used for only a single application).

- A URI-ID is direct (provided by a user) and restricted (can be used for only a single application).

We summarize this taxonomy in the following table.

	Direct	Restricted
CN-ID	Yes	No
DNS-ID	Yes	No

SRV-ID	Either	Yes
URI-ID	Yes	Yes

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 a DNS domain name that conforms to one of the following forms:

1. A "traditional domain name", i.e., a fully-qualified domain name or "FQDN" (see [DNS-CONCEPTS]) all of whose labels are "LDH labels" as defined 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 germane

to this specification.

2. An "internationalized domain name", i.e., a DNS domain name that conforms to the overall form of a domain name (dot-separated labels) but that can include Unicode code points outside the traditional US-ASCII range or, more precisely, either U-labels or A-labels as described in [IDNA-DEFS] and the associated documents.

2.3. Subject Naming in PKIX Certificates

In theory, the Internet Public Key Infrastructure using X.509 [PKIX] employs the global directory service model defined in [X.500] and [X.501]. In that model, information is held in a directory information base (DIB) and entries in the DIB are organized in a hierarchy called the directory information tree (DIT). An object or alias entry in that hierarchy consists of a set of attributes (each of which has a defined type and one or more values) and is uniquely identified by a Distinguished Name (DN). The DN of an entry is constructed by combining the Relative Distinguished Names of its superior entries in the tree (all the way down to the root of the DIT) with one or more specially-nominated attributes of the entry itself (which together comprise the Relative Distinguished Name (RDN) of the entry, so-called because it is relative to the Distinguished Names of the superior entries in the tree). The entry closest to the root is sometimes referred to as the "most significant" entry and the entry farthest from the root is sometimes referred to as the "least significant" entry. An RDN is a set (i.e., an unordered group) of attribute-type-and-value pairs (see also [LDAP-DN]), each of which asserts some attribute about the entry.

In practice, the certificates used in [X.509] and [PKIX] borrow key concepts of X.500 and X.501 (e.g., DNs and RDNs) to identify entities, but such certificates are not necessarily part of a global directory information base. Specifically, the subject field of a PKIX certificate is an X.501 type Name that "identifies the entity associated with the public key stored in the subject public key field" (see Section 4.1.2.6 of [PKIX]). However, it is perfectly acceptable for the subject field to be empty, as long as the certificate contains a subjectAltName extension that includes at least one subjectAltName entry, because the subject alternative name ("subjectAltName") extension allows various identities to be bound to the subject (see Section 4.2.1.6 of [PKIX]). The subjectAltName

extension itself is a sequence of typed entries, where each type is a distinct kind of identifier.

For our purposes, an application service is identified by a name or names carried in the subject field and/or in one of the following identifier types:

- DNS-ID
- SRV-ID
- URI-ID

Existing certificates often use a CN-ID in the subject field to represent a fully-qualified DNS domain name; for example, consider the following subject name, where the attribute of type Common Name contains a string whose form matches that of a fully-qualified DNS domain name of "www.example.com":

CN=www.example.com,C=GB,OU=Web Services

In general, this specification recommends and prefers use of subjectAltName entries (DNS-ID, SRV-ID, URI-ID, etc.) over use of the subject field (CN-ID) where possible, as more completely described in the following sections. However, profiles of this specification can legitimately encourage continued support for the CN-ID identifier type if they have good reasons to do so, such as backward compatibility with deployed infrastructure.

> Implementation Note: Confusion sometimes arises from different renderings or encodings of the hierarchical information contained in a certificate. Certificates are binary objects and are encoded using the Distinguished Encoding Rules (DER) specified in [X.690]. However, some implementations generate displayable (a.k.a. printable) renderings of the certificate issuer, subject field, and subjectAltName extension, and these renderings convert the DER-encoded sequences into a "string representation" before being displayed. Because a Distinguished Name (DN) is an ordered sequence, order is typically preserved in the DN string representations, although the two most prevalent DN string representations differ in employing left-to-right vs. right-to-left ordering. However, because a Relative Distinguished Name (RDN) is an unordered group of attribute-type-and-value pairs, the string representation of an RDN can differ from the canonical DER encoding (and the order of attribute-type-and-value pairs can differ in the RDN string representations or display orders provided by various implementations). Furthermore, various specifications refer to the order of RDNs using terminology that is not directly related to the information hierarchy, such as

"most specific" vs. "least specific", "left-most" vs. "right-

most", "first" vs. "last", or "most significant" vs. "least significant" (see for example [LDAP-DN]). To reduce confusion, in this specification we avoid such terms and instead use the terms provided under Section 1.5; in particular, we do not use the term "(most specific) Common Name field in the Subject field" from [HTTP-TLS] and instead state that a CN-ID is a Relative Distinguished Name (RDN) in the certificate subject that contains one and only one attribute-type-and-value pair of type Common Name (thus removing the possibility that an RDN might contain multiple AVAs of type CN, one of which would be considered "most specific").

3. Representation of Server Identity

3.1. 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.

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 in which a server is discovered by means of DNS SRV records [DNS-SRV] (e.g., this is true of [XMPP]), then the certificate SHOULD include an "SRV-ID"; furthermore, multiple instances of the "SRV-ID" identifier type are allowed.

3. If the service using the certificate deploys a technology in which a server is typically associated with a URI (e.g., this is true of [SIP]), then the certificate SHOULD include a URI-ID; the scheme SHALL be that of the protocol associated with the service type and the authority component SHALL be the fully-qualified DNS domain name of the service; furthermore, multiple instances of the "URI-ID" identifier type are allowed.

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. Even though many deployed clients still check for the CN-ID within the certificate subject field, the certificate SHOULD NOT represent the server's fully-qualified DNS domain name in a CN-ID unless a profile of this specification encourages continued support for the CN-ID identifier type.

6. The certificate MAY contain more than one DNS-ID, but SHOULD NOT contain more than one CN-ID.

7. The fully-qualified DNS domain name portion of a presented identifier SHOULD NOT contain the wildcard character '*', whether as the complete left-most label within the identifier (following the definition of "label" from [DNS], e.g., "*.example.com") or as a fragment thereof (e.g., *oo.example.com, f*o.example.com, or foo*.example.com). For details, see Section 5.2.

3.2. Examples

A certificate for the website at "www.example.com" might include only a DNS-ID of "www.example.com" (and, strictly as a fallback for existing client software, a CN-ID of "www.example.com").

A certificate for the IMAP-accessible email server at "mail.example.com" might include SRV-IDs of "_imap.mail.example.com" and "_imaps.mail.example.com" (see [EMAIL-SRV]) and a DNS-ID of "mail.example.com".

A certificate for the XMPP-compatible instant messaging server at "im.example.com" might include SRV-IDs of "_xmpp-client.im.example.com" and "_xmpp-server.im.example.com" (see [XMPP]), a DNS-ID of "im.example.com", and an XMPP-specific "XmppAddr" of "im.example.com" (see [XMPP]).

4. Verification of Service Identity

4.1. 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 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.

4.2. Constructing a List of Reference Identifiers

4.2.1. 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 HTTP URL for a web site), 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 server portion of the user's account name), 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 a service type.

The client might need to extract the source domain and 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., extracting the fully-qualified DNS domain name from the authority component of a URI or extracting the service type from the scheme of a URI) or information for which the extraction is performed in a manner that is not subject to subversion by network attackers (e.g., pulling the data from a delegated domain that 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 that provides both mutual authentication and integrity checking).

For an interactive client, it is strongly encouraged that each reference identifier SHOULD be based on the source domain provided by the user 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 connection. There is only one scenario in which it is acceptable for an interactive client to override the recommendation in this rule and therefore connect to a domain name other than the source domain (e.g., to a derived domain or some other domain name that is presented by the server in a certificate): because a human user has "pinned" the application service's certificate to the alternative domain name as further discussed under Section 4.6.4 and Section 5.1. 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 service type, the client constructs a list of reference identifiers in accordance with the following rules:

- The list MUST 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 service type is typically discovered by means of DNS SRV records, then the list SHOULD include an SRV-ID.

- If a server for the service type is typically associated with a URI, then the list SHOULD include a URI-ID.

- The list MAY include a CN-ID, mainly for the sake of backward compatibility with existing infrastructure.

The client does not need to construct the foregoing identifiers in the actual formats found in a certificate (e.g., as ASN.1 types), it only needs to construct the functional equivalent of such identifiers for matching purposes.

> Security Note: A client MUST NOT construct a reference identifier corresponding to Relative Distinguished Names (RDNs) other than those of type Common Name and MUST NOT check for RDNs other than those of type Common Name in the presented identifiers.

4.2.2. Examples

A web browser that is connecting via HTTP to the website at "www.example.com" might have two reference identifiers: a DNS-ID of "www.example.com" and, as a fallback, a CN-ID of "www.example.com".

A mail user agent that is connecting via IMAP to the email service at "mail.example.com" might have two reference identifiers: an SRV-ID of "_imaps.mail.example.com" (see [EMAIL-SRV]) and a DNS-ID of "mail.example.com".

An instant messaging (IM) client that is connecting via XMPP to the

IM service at "im.example.com" might have three reference identifiers: an SRV-ID of "_xmpp.im.example.com", a DNS-ID of "im.example.com", and an XMPP-specific "XmppAddr" of "im.example.com" (see [XMPP]).

4.3. Seeking 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.

> Implementation Note: A client might be configured to perform multiple searches, i.e., to match more than one reference identifier; although such behavior is not forbidden by this document, rules for matching multiple reference identifiers are a matter for implementation or future specification.
>
> Security Note: A client MUST NOT seek a match for a reference identifier of CN-ID if the presented identifiers include a DNS-ID, SRV-ID, URI-ID, or any application-specific identifier types supported by the client.

Detailed comparison rules for finding a match are provided in the following sections.

4.4. Verifying a Domain Name

The client MUST match the DNS domain name portion of a reference identifier according to the following rules (and SHOULD also check the service type as described under Section 4.5). 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, in case the client supports presented identifiers that contain the wildcard character '*', we define a supplemental rule for so-called "wildcard domains". We also specify the circumstances under which it is acceptable to check the "CN-ID"

identifier type.

4.4.1. 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 components 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 4.4.3).

4.4.2. 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 every label in the domain name to an A-label (as described in [IDNA-DEFS]) before checking the domain name. 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 4.4.3).

4.4.3. Checking of Wildcard Labels

A client employing this specification's rules MAY match the reference identifier against a presented identifier whose DNS domain name portion contains the wildcard character '*' as part or all of a label (following the definition of "label" from [DNS]). For information regarding the security characteristics of wildcard certificates, see Section 5.2.

If a client matches the reference identifier against a presented identifier whose DNS domain name portion contains the wildcard character '*', the following rules apply:

1. The client SHOULD NOT attempt to match a presented identifier in which the wildcard character comprises a label other than the left-most label (e.g., do not match bar.*.example.net).

2. If the wildcard character is the only character of the left-most label in the presented identifier, the client SHOULD NOT compare against anything but the left-most label of the reference identifier (e.g., *.example.com would match foo.example.com but not bar.foo.example.com or example.com).

3. The client MAY match a presented identifier in which the wildcard character is not the only character of the label (e.g., baz*.example.net and *baz.example.net and b*z.example.net would be taken to match baz1.example.net and foobaz.example.net and buzz.example.net, respectively).

4.4.4. Checking of Common Names

As noted, a client MUST NOT seek a match for a reference identifier of CN-ID if the presented identifiers include a DNS-ID, SRV-ID, URI-ID, or any application-specific identifier types supported by the client.

Therefore, if and only if the presented identifiers do not include a DNS-ID, SRV-ID, URI-ID, or any application-specific identifier types supported by the client, then the client MAY as a last resort check for a string whose form matches that of a fully-qualified DNS domain name in the CN-ID. If the client chooses to compare a reference identifier of type CN-ID against that string, it MUST follow the comparison rules for the DNS domain name portion of an identifier of type DNS-ID, SRV-ID, or URI-ID, as described under Section 4.4.1, Section 4.4.2, and Section 4.4.3.

4.5. Verifying an Application Type

When checking identifiers of type SRV-ID and URI-ID, a client SHOULD check not only the domain name but also the service type of the service to which it connects. This is a best practice because typically a client is not designed to connect to all kinds of services using all possible application protocols, but instead is designed to connect to one kind of service, such as a web site, an email service, or an instant messaging service.

The service type is verified by means of either an SRV-ID or URI-ID.

4.5.1. SRV-ID

The service name portion of an SRV-ID (e.g., "xmpp") 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]).

4.5.2. 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 therefore does not need to be included in any comparison.

4.6. Outcome

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

4.6.1. Case #1: Match Found

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

4.6.2. 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 connection attempt (as "pinning" is explained under Section 1.5), and the presented certificate matches the pinned certificate, then the service identity check succeeds.

4.6.3. 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 connection attempt, then the client MUST proceed as described under Section 4.6.4.

4.6.4. 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 connection with a bad certificate

error; this behavior is preferable because it prevents users from inadvertently bypassing security protections in hostile situations.

Security Note: Some interactive clients give advanced users the option of proceeding with acceptance despite the identity mismatch, thereby "pinning" the certificate to one of the

reference identifiers in the list constructed by the client for this connection attempt. Although this behavior can be appropriate in certain specialized circumstances, in general it ought to be exposed only to advanced users. Even then it needs to be handled with extreme caution, for example by first encouraging even an advanced user to terminate the connection and, if the advanced user chooses to proceed anyway, by forcing the user to view the entire certification path and only then allowing the user to pin the certificate (on a temporary or permanent basis, at the user's option).

Otherwise, if the client is an automated application not directly controlled by a human user, then it SHOULD terminate the connection 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.

5. Security Considerations

5.1. Pinned Certificates

As defined under Section 1.5, 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 the that 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 service type, the service's derived domain and port number, and other relevant information such as the subject field's Common Name, Organization, and Country).

5.2. Wildcard Certificates

This document states that the wildcard character '*' SHOULD NOT be included in presented identifiers but MAY be checked by application clients (mainly for the sake of backward compatibility with existing infrastructure); as a result, the rules provided in this document are more restrictive than the rules for many existing application technologies (see Appendix A). Several security considerations justify tightening the rules:

- The inclusion of the wildcard character in certificates has led to homograph attacks involving non-ASCII characters that look similar to characters commonly included in HTTP URLs, such as "/" and "?";

  for discussion, see for example [Defeating-SSL] (beginning at slide 91).

- The ability to obtain a certificate containing the wildcard character might broaden the range of applications services that an attacker could forge, thus increasing (a) the chances that attackers would attempt to steal credentials needed for obtaining certificates and (b) the potential damage that would result from a successful attack.

- Specifications for existing application technologies are not clear about whether the wildcard character is allowed only as the complete left-most label (e.g., *.example.com), some fragment of the left-most label (e.g., foo*.example.com, f*o.example.com, or *oo.example.com), or even all or part of a label other than the left-most label (e.g., www.*.example.com or www.foo*.example.com); nor are they clear about whether a presented identifier can include more than one instance of the wildcard character (e.g., f*b*r.example.com or *.*.example.com). These ambiguities might introduce exploitable differences in identity checking behavior among client implementations and necessitate overly complex and inefficient identity checking algorithms.

Notwithstanding the foregoing security considerations, profiles of this specification can legitimately encourage continued support for the wildcard character if they have good reasons to do so, such as backward compatibility with deployed infrastructure.

5.3. 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].

6. IANA Considerations

This document specifies no actions for the IANA.

7. References

7.1. Normative References

[DNS] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[DNS-CONCEPTS] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[DNS-SRV] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[IDNA-DEFS] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[LDAP-DN] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, June 2006.

[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, May 2008.

[SRVNAME] Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, August 2007.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

7.2. Informative References

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

[DNS-CASE]
Eastlake, D., "Domain Name System (DNS) Case Insensitivity Clarification", RFC 4343, January 2006.

[DNSSEC] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[DTLS] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[EMAIL-SRV]
Daboo, C., "Use of SRV Records for Locating Email Submission/Access services", draft-daboo-srv-email-05 (work in progress), May 2010.

[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>.

[GIST] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, October 2010.

[HTTP] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[HTTP-TLS] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[IMAP] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[IDNA2003] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[IP] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[LDAP] Sermersheim, J., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[LDAP-AUTH] Harrison, R., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms", RFC 4513, June 2006.

[LDAP-SCHEMA] Sciberras, A., "Lightweight Directory Access Protocol (LDAP): Schema for User Applications", RFC 4519, June 2006.

[LDAP-TLS] Hodges, J., Morgan, R., and M. Wahl, "Lightweight

Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.

[NAPTR] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[NETCONF] Enns, R., "NETCONF Configuration Protocol", RFC 4741, December 2006.

[NETCONF-SSH] Wasserman, M. and T. Goddard, "Using the NETCONF Configuration Protocol over Secure SHell (SSH)", RFC 4742, December 2006.

[NETCONF-TLS] Badra, M., "NETCONF over Transport Layer Security (TLS)", RFC 5539, May 2009.

[NNTP] Feather, C., "Network News Transfer Protocol (NNTP)", RFC 3977, October 2006.

[NNTP-TLS] Murchison, K., Vinocur, J., and C. Newman, "Using Transport Layer Security (TLS) with Network News Transfer Protocol (NNTP)", RFC 4642, October 2006.

[OCSP] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.

[OPENPGP] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[POP3] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[PRIVATE] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[PKIX-OLD]
Housley, R., Ford, W., Polk, T., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[S-NAPTR] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[SECTERMS]
Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[SIP-CERTS]
Gurbani, V., Lawrence, S., and A. Jeffrey, "Domain Certificates in the Session Initiation Protocol (SIP)", RFC 5922, June 2010.

[SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[SMTP-AUTH]
Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[SMTP-TLS]
Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[SNMP] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[SNMP-TLS]
Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5953, August 2010.

[SYSLOG] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[SYSLOG-DTLS]
Salowey, J., Petch, T., Gerhards, R., and H. Feng,

"Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog", RFC 6012, October 2010.

[SYSLOG-TLS] Miao, F., Ma, Y., and J. Salowey, "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, March 2009.

[TLS] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

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

[USINGTLS] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[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>.

[X.500] International Telecommunications Union, "Information Technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services", ITU-T Recommendation X.500, ISO Standard 9594-1, August 2005.

[X.501] International Telecommunications Union, "Information Technology - Open Systems Interconnection - The Directory: Models", ITU-T Recommendation X.501, ISO Standard 9594-2, August 2005.

[X.509] International Telecommunications Union, "Information Technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, ISO Standard 9594-8, August 2005.

[X.520] International Telecommunications Union, "Information Technology - Open Systems Interconnection - The Directory: Selected attribute types", ITU-T Recommendation X.509,

ISO Standard 9594-6, August 2005.

[X.690] International Telecommunications Union, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and

Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO Standard 8825-1, August 2008.

[XMPP] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-xmpp-3920bis-17 (work in progress), October 2010.

[XMPP-OLD] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.

Appendix A. Prior Art

(This section is non-normative.)

The recommendations in this document are an abstraction from recommendations in specifications for a wide range of application protocols. For the purpose of comparison and to delineate the history of thinking about application service identity verification within the IETF, this informative section gathers together prior art by including the exact text from various RFCs (the only modifications are changes to the names of several references to maintain coherence with the main body of this document, and the elision of irrelevant text as marked by the characters "[...]").

A.1. IMAP, POP3, and ACAP (1999)

In 1999, [USINGTLS] specified the following text regarding application service identity verification in IMAP, POP3, and ACAP:

######

2.4. Server Identity Check

During the TLS negotiation, the client MUST check its understanding

of the server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks. Matching is performed according to these rules:

- The client MUST use the server hostname it used to open the connection as the value to compare against the server name as expressed in the server certificate. The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.
- If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity.

- Matching is case-insensitive.
- A "*" wildcard character MAY be used as the left-most name component in the certificate. For example, *.example.com would match a.example.com, foo.example.com, etc. but would not match example.com.
- If the certificate contains multiple names (e.g. more than one dNSName field), then a match with any one of the fields is considered acceptable.

If the match fails, the client SHOULD either ask for explicit user confirmation, or terminate the connection and indicate the server's identity is suspect.

######

A.2. HTTP (2000)

In 2000, [HTTP-TLS] specified the following text regarding application service identity verification in HTTP:

######

3.1. Server Identity

In general, HTTP/TLS requests are generated by dereferencing a URI. As a consequence, the hostname for the server is known to the client. If the hostname is available, the client MUST check it against the server's identity as presented in the server's Certificate message, in order to prevent man-in-the-middle attacks.

If the client has external information as to the expected identity of the server, the hostname check MAY be omitted. (For instance, a client may be connecting to a machine whose address and hostname are dynamic but the client knows the certificate that the server will present.) In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man in the middle attacks. In special cases, it may be appropriate for the client to simply ignore the server's identity, but it must be understood that this leaves the connection open to active attack.

If a subjectAltName extension of type dNSName is present, that MUST be used as the identity. Otherwise, the (most specific) Common Name field in the Subject field of the certificate MUST be used. Although the use of the Common Name is existing practice, it is deprecated and Certification Authorities are encouraged to use the dNSName instead.

Matching is performed using the matching rules specified by [PKIX-OLD]. If more than one identity of a given type is present in

the certificate (e.g., more than one dNSName name, a match in any one of the set is considered acceptable.) Names may contain the wildcard character * which is considered to match any single domain name component or component fragment. E.g., *.a.com matches foo.a.com but not bar.foo.a.com. f*.com matches foo.com but not bar.com.

In some cases, the URI is specified as an IP address rather than a hostname. In this case, the iPAddress subjectAltName must be present in the certificate and must exactly match the IP in the URI.

If the hostname does not match the identity in the certificate, user oriented clients MUST either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error. Automated clients MUST log the error to an appropriate audit log (if available) and SHOULD terminate the connection (with a bad certificate error). Automated clients MAY provide a configuration setting that disables this check, but MUST provide a setting which enables it.

Note that in many cases the URI itself comes from an untrusted source. The above-described check provides no protection against attacks where this source is compromised. For example, if the URI

was obtained by clicking on an HTML page which was itself obtained without using HTTP/TLS, a man in the middle could have replaced the URI. In order to prevent this form of attack, users should carefully examine the certificate presented by the server to determine if it meets their expectations.

######

A.3. LDAP (2000/2006)

In 2000, [LDAP-TLS] specified the following text regarding application service identity verification in LDAP:

######

3.6. Server Identity Check

The client MUST check its understanding of the server's hostname against the server's identity as presented in the server's Certificate message, in order to prevent man-in-the-middle attacks.

Matching is performed according to these rules:

- The client MUST use the server hostname it used to open the LDAP connection as the value to compare against the server name as expressed in the server's certificate. The client MUST NOT use

the server's canonical DNS name or any other derived form of name.
- If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity.
- Matching is case-insensitive.
- The "*" wildcard character is allowed. If present, it applies only to the left-most name component.

E.g. *.bar.com would match a.bar.com, b.bar.com, etc. but not bar.com. If more than one identity of a given type is present in the certificate (e.g. more than one dNSName name), a match in any one of the set is considered acceptable.

If the hostname does not match the dNSName-based identity in the certificate per the above check, user-oriented clients SHOULD either

notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection and indicate that the server's identity is suspect. Automated clients SHOULD close the connection, returning and/or logging an error indicating that the server's identity is suspect.

Beyond the server identity checks described in this section, clients SHOULD be prepared to do further checking to ensure that the server is authorized to provide the service it is observed to provide. The client MAY need to make use of local policy information.

######

In 2006, [LDAP-AUTH] specified the following text regarding application service identity verification in LDAP:

######

3.1.3. Server Identity Check

In order to prevent man-in-the-middle attacks, the client MUST verify the server's identity (as presented in the server's Certificate message). In this section, the client's understanding of the server's identity (typically the identity used to establish the transport connection) is called the "reference identity".

The client determines the type (e.g., DNS name or IP address) of the reference identity and performs a comparison between the reference identity and each subjectAltName value of the corresponding type until a match is produced. Once a match is produced, the server's identity has been verified, and the server identity check is complete. Different subjectAltName types are matched in different ways. Sections 3.1.3.1 - 3.1.3.3 explain how to compare values of

various subjectAltName types.

The client may map the reference identity to a different type prior to performing a comparison. Mappings may be performed for all available subjectAltName types to which the reference identity can be mapped; however, the reference identity should only be mapped to types for which the mapping is either inherently secure (e.g., extracting the DNS name from a URI to compare with a subjectAltName

of type dNSName) or for which the mapping is performed in a secure manner (e.g., using [DNSSEC], or using user- or admin-configured host-to-address/address-to-host lookup tables).

The server's identity may also be verified by comparing the reference identity to the Common Name (CN) [LDAP-SCHEMA] value in the last Relative Distinguished Name (RDN) of the subject field of the server's certificate (where "last" refers to the DER-encoded order, not the order of presentation in a string representation of DER-encoded data). This comparison is performed using the rules for comparison of DNS names in Section 3.1.3.1, below, with the exception that no wildcard matching is allowed. Although the use of the Common Name value is existing practice, it is deprecated, and Certification Authorities are encouraged to provide subjectAltName values instead. Note that the TLS implementation may represent DNs in certificates according to X.500 or other conventions. For example, some X.500 implementations order the RDNs in a DN using a left-to-right (most significant to least significant) convention instead of LDAP's right-to-left convention.

If the server identity check fails, user-oriented clients SHOULD either notify the user (clients may give the user the opportunity to continue with the LDAP session in this case) or close the transport connection and indicate that the server's identity is suspect. Automated clients SHOULD close the transport connection and then return or log an error indicating that the server's identity is suspect or both.

Beyond the server identity check described in this section, clients should be prepared to do further checking to ensure that the server is authorized to provide the service it is requested to provide. The client may need to make use of local policy information in making this determination.

3.1.3.1. Comparison of DNS Names

If the reference identity is an internationalized domain name, conforming implementations MUST convert it to the ASCII Compatible Encoding (ACE) format as specified in Section 4 of RFC 3490 [IDNA2003] before comparison with subjectAltName values of type

dNSName. Specifically, conforming implementations MUST perform the

conversion operation specified in Section 4 of RFC 3490 as follows:

- in step 1, the domain name SHALL be considered a "stored string";
- in step 3, set the flag called "UseSTD3ASCIIRules";
- in step 4, process each label with the "ToASCII" operation; and
- in step 5, change all label separators to U+002E (full stop).

After performing the "to-ASCII" conversion, the DNS labels and names MUST be compared for equality according to the rules specified in Section 3 of RFC3490.

The '*' (ASCII 42) wildcard character is allowed in subjectAltName values of type dNSName, and then only as the left-most (least significant) DNS label in that value. This wildcard matches any left-most DNS label in the server name. That is, the subject *.example.com matches the server names a.example.com and b.example.com, but does not match example.com or a.b.example.com.

3.1.3.2. Comparison of IP Addresses

When the reference identity is an IP address, the identity MUST be converted to the "network byte order" octet string representation [IP] [IPv6]. For IP Version 4, as specified in RFC 791, the octet string will contain exactly four octets. For IP Version 6, as specified in RFC 2460, the octet string will contain exactly sixteen octets. This octet string is then compared against subjectAltName values of type iPAddress. A match occurs if the reference identity octet string and value octet strings are identical.

3.1.3.3. Comparison of Other subjectName Types

Client implementations MAY support matching against subjectAltName values of other types as described in other documents.

######

A.4. SMTP (2002/2007)

In 2002, [SMTP-TLS] specified the following text regarding application service identity verification in SMTP:

######

4.1 Processing After the STARTTLS Command

[...]

The decision of whether or not to believe the authenticity of the other party in a TLS negotiation is a local matter. However, some general rules for the decisions are:

- o A SMTP client would probably only want to authenticate an SMTP server whose server certificate has a domain name that is the domain name that the client thought it was connecting to.

[...]

######

In 2006, [SMTP-AUTH] specified the following text regarding application service identity verification in SMTP:

######

14. Additional Requirements When Using SASL PLAIN over TLS

[...]

After a successful [TLS] negotiation, the client MUST check its understanding of the server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks. If the match fails, the client MUST NOT attempt to authenticate using the SASL PLAIN mechanism. Matching is performed according to the following rules:

> The client MUST use the server hostname it used to open the connection as the value to compare against the server name as expressed in the server certificate. The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.
> If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity.
> Matching is case-insensitive.
> A "*" wildcard character MAY be used as the leftmost name component in the certificate. For example, *.example.com would match a.example.com, foo.example.com, etc., but would not match example.com.
> If the certificate contains multiple names (e.g., more than one dNSName field), then a match with any one of the fields is considered acceptable.

######

A.5. XMPP (2004)

In 2004, [XMPP-OLD] specified the following text regarding application service identity verification in XMPP:

######

14.2. Certificate Validation

When an XMPP peer communicates with another peer securely, it MUST validate the peer's certificate. There are three possible cases:

Case #1: The peer contains an End Entity certificate which appears to be certified by a certification path terminating in a trust anchor (as described in Section 6.1 of [PKIX]).
Case #2: The peer certificate is certified by a Certificate Authority not known to the validating peer.
Case #3: The peer certificate is self-signed.

In Case #1, the validating peer MUST do one of two things:

1. Verify the peer certificate according to the rules of [PKIX]. The certificate SHOULD then be checked against the expected identity of the peer following the rules described in [HTTP-TLS], except that a subjectAltName extension of type "xmpp" MUST be used as the identity if present. If one of these checks fails, user-oriented clients MUST either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error. Automated clients SHOULD terminate the connection (with a bad certificate error) and log the error to an appropriate audit log. Automated clients MAY provide a configuration setting that disables this check, but MUST provide a setting that enables it.
2. The peer SHOULD show the certificate to a user for approval, including the entire certification path. The peer MUST cache the certificate (or some non-forgeable representation such as a hash). In future connections, the peer MUST verify that the same certificate was presented and MUST notify the user if it has changed.

In Case #2 and Case #3, implementations SHOULD act as in (2) above.

######

At the time of this writing, [XMPP] refers to this document for rules regarding application service identity verification in XMPP.

A.6. NNTP (2006)

In 2006, [NNTP-TLS] specified the following text regarding application service identity verification in NNTP:

######

5. Security Considerations

[...]

During the TLS negotiation, the client MUST check its understanding of the server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks. Matching is performed according to these rules:

- The client MUST use the server hostname it used to open the connection (or the hostname specified in TLS "server_name" extension [TLS]) as the value to compare against the server name as expressed in the server certificate. The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.
- If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity.
- Matching is case-insensitive.
- A "*" wildcard character MAY be used as the left-most name component in the certificate. For example, *.example.com would match a.example.com, foo.example.com, etc., but would not match example.com.
- If the certificate contains multiple names (e.g., more than one

dNSName field), then a match with any one of the fields is considered acceptable.

If the match fails, the client SHOULD either ask for explicit user confirmation or terminate the connection with a QUIT command and indicate the server's identity is suspect.

Additionally, clients MUST verify the binding between the identity of the servers to which they connect and the public keys presented by those servers. Clients SHOULD implement the algorithm in Section 6 of [PKIX] for general certificate validation, but MAY supplement that algorithm with other validation methods that achieve equivalent levels of verification (such as comparing the server certificate against a local store of already-verified certificates and identity bindings).

######

A.7. NETCONF (2006/2009)

In 2006, [NETCONF-SSH] specified the following text regarding application service identity verification in NETCONF:

######

6. Security Considerations

The identity of the server MUST be verified and authenticated by the client according to local policy before password-based authentication data or any configuration or state data is sent to or received from the server. The identity of the client MUST also be verified and authenticated by the server according to local policy to ensure that the incoming client request is legitimate before any configuration or state data is sent to or received from the client. Neither side should establish a NETCONF over SSH connection with an unknown, unexpected, or incorrect identity on the opposite side.

######

In 2009, [NETCONF-TLS] specified the following text regarding application service identity verification in NETCONF:

######

3.1. Server Identity

During the TLS negotiation, the client MUST carefully examine the certificate presented by the server to determine if it meets the client's expectations. Particularly, the client MUST check its understanding of the server hostname against the server's identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks.

Matching is performed according to the rules below (following the example of [NNTP-TLS]):

- The client MUST use the server hostname it used to open the connection (or the hostname specified in the TLS "server_name" extension [TLS]) as the value to compare against the server name as expressed in the server certificate. The client MUST NOT use any form of the server hostname derived from an insecure remote source (e.g., insecure DNS lookup). CNAME canonicalization is not done.

- If a subjectAltName extension of type dNSName is present in the certificate, it MUST be used as the source of the server's identity.
- Matching is case-insensitive.
- A "*" wildcard character MAY be used as the leftmost name component in the certificate. For example, *.example.com would match a.example.com, foo.example.com, etc., but would not match example.com.
- If the certificate contains multiple names (e.g., more than one dNSName field), then a match with any one of the fields is considered acceptable.

If the match fails, the client MUST either ask for explicit user confirmation or terminate the connection and indicate the server's identity is suspect.

Additionally, clients MUST verify the binding between the identity of the servers to which they connect and the public keys presented by

those servers. Clients SHOULD implement the algorithm in Section 6 of [PKIX] for general certificate validation, but MAY supplement that algorithm with other validation methods that achieve equivalent levels of verification (such as comparing the server certificate against a local store of already-verified certificates and identity bindings).

If the client has external information as to the expected identity of the server, the hostname check MAY be omitted.

######

A.8. Syslog (2009)

In 2009, [SYSLOG-TLS] specified the following text regarding application service identity verification in Syslog:

######

5.2. Subject Name Authorization

Implementations MUST support certification path validation [PKIX]. In addition, they MUST support specifying the authorized peers using locally configured host names and matching the name against the certificate as follows.

- Implementations MUST support matching the locally configured host name against a dNSName in the subjectAltName extension field and SHOULD support checking the name against the common name portion of the subject distinguished name.

- The '*' (ASCII 42) wildcard character is allowed in the dNSName of the subjectAltName extension (and in common name, if used to store the host name), but only as the left-most (least significant) DNS label in that value. This wildcard matches any left-most DNS label in the server name. That is, the subject *.example.com matches the server names a.example.com and b.example.com, but does not match example.com or a.b.example.com. Implementations MUST support wildcards in certificates as specified above, but MAY provide a configuration option to disable them.
- Locally configured names MAY contain the wildcard character to match a range of values. The types of wildcards supported MAY be

more flexible than those allowed in subject names, making it possible to support various policies for different environments. For example, a policy could allow for a trust-root-based authorization where all credentials issued by a particular CA trust root are authorized.

- o If the locally configured name is an internationalized domain name, conforming implementations MUST convert it to the ASCII Compatible Encoding (ACE) format for performing comparisons, as specified in Section 7 of [PKIX].
- o Implementations MAY support matching a locally configured IP address against an iPAddress stored in the subjectAltName extension. In this case, the locally configured IP address is converted to an octet string as specified in [PKIX], Section 4.2.1.6. A match occurs if this octet string is equal to the value of iPAddress in the subjectAltName extension.

######

A.9. SIP (2010)

In 2010, [SIP-CERTS] specified the following text regarding application service identity verification in SIP:

######

7.2. Comparing SIP Identities

When an implementation (either client or server) compares two values as SIP domain identities:

Implementations MUST compare only the DNS name component of each SIP domain identifier; an implementation MUST NOT use any scheme or parameters in the comparison.

Implementations MUST compare the values as DNS names, which means that the comparison is case insensitive as specified by [DNS-CASE]. Implementations MUST handle Internationalized Domain Names (IDNs) in accordance with Section 7.2 of [PKIX].

Implementations MUST match the values in their entirety:

Implementations MUST NOT match suffixes. For example, "foo.example.com" does not match "example.com".

Implemenations MUST NOT match any form of wildcard, such as a

leading "." or "*." with any other DNS label or sequence of labels. For example, "*.example.com" matches only "*.example.com" but not "foo.example.com". Similarly, ".example.com" matches only ".example.com", and does not match "foo.example.com."

[HTTP-TLS] allows the dNSName component to contain a wildcard; e.g., "DNS:*.example.com". [PKIX], while not disallowing this explicitly, leaves the interpretation of wildcards to the individual specification. [SIP] does not provide any guidelines on the presence of wildcards in certificates. Through the rule above, this document prohibits such wildcards in certificates for SIP domains.

######

A.10. SNMP (2010)

In 2010, [SNMP-TLS] specified the following text regarding application service identity verification in SNMP:

######

If the server's presented certificate has passed certification path validation [PKIX] to a configured trust anchor, and an active row exists with a zero-length snmpTlstmAddrServerFingerprint value, then the snmpTlstmAddrServerIdentity column contains the expected host name. This expected host name is then compared against the server's certificate as follows:

- Implementations MUST support matching the expected host name against a dNSName in the subjectAltName extension field and MAY support checking the name against the CommonName portion of the subject distinguished name.

- The '*' (ASCII 0x2a) wildcard character is allowed in the dNSName of the subjectAltName extension (and in common name, if used to store the host name), but only as the left-most (least significant) DNS label in that value. This wildcard matches any left-most DNS label in the server name. That is, the subject *.example.com matches the server names a.example.com and b.example.com, but does not match example.com or a.b.example.com. Implementations MUST support wildcards in certificates as specified above, but MAY provide a configuration option to disable them.

- o If the locally configured name is an internationalized domain name, conforming implementations MUST convert it to the ASCII Compatible Encoding (ACE) format for performing comparisons, as specified in Section 7 of [PKIX].

If the expected host name fails these conditions then the connection MUST be closed.

######

A.11. GIST (2010)

In 2010, [GIST] specified the following text regarding application service identity verification in the General Internet Signalling Transport:

######

5.7.3.1. Identity Checking in TLS

After TLS authentication, a node MUST check the identity presented by the peer in order to avoid man-in-the-middle attacks, and verify that the peer is authorised to take part in signalling at the GIST layer. The authorisation check is carried out by comparing the presented identity with each Authorised Peer Database (APD) entry in turn, as discussed in Section 4.4.2. This section defines the identity comparison algorithm for a single APD entry.

For TLS authentication with X.509 certificates, an identity from the DNS namespace MUST be checked against each subjectAltName extension of type dNSName present in the certificate. If no such extension is present, then the identity MUST be compared to the (most specific) Common Name in the Subject field of the certificate. When matching DNS names against dNSName or Common Name fields, matching is case-insensitive. Also, a "*" wildcard character MAY be used as the left-most name component in the certificate or identity in the APD. For example, *.example.com in the APD would match certificates for a.example.com, foo.example.com, *.example.com, etc., but would not match example.com. Similarly, a certificate for *.example.com would be valid for APD identities of a.example.com, foo.example.com, *.example.com, etc., but not example.com.

Additionally, a node MUST verify the binding between the identity of the peer to which it connects and the public key presented by that peer. Nodes SHOULD implement the algorithm in Section 6 of [PKIX] for general certificate validation, but MAY supplement that algorithm with other validation methods that achieve equivalent levels of

verification (such as comparing the server certificate against a

local store of already-verified certificates and identity bindings).

For TLS authentication with pre-shared keys, the identity in the psk_identity_hint (for the server identity, i.e. the Responding node) or psk_identity (for the client identity, i.e. the Querying node) MUST be compared to the identities in the APD.

######

Authors' Addresses

Peter Saint-Andre
Cisco
1899 Wyknoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Jeff Hodges
PayPal
2211 North First Street
San Jose, California 95131
US

Email: Jeff.Hodges@PayPal.com