Network Working Group
Internet-Draft
Intended status: BCP
Expires: November 1, 2010

P. Saint-Andre, Ed.
Cisco
J. Hodges, Ed.
PayPal

April 30, 2010

Representation and Verification of Application Server Identity in Certificates Used with Transport Layer Security (TLS)
draft-saintandre-tls-server-id-check-04

Abstract

Many application technologies enable a secure connection between two entities using certificates in the context of Transport Layer Security (TLS). This document specifies best current practices for representing and verifying the identity of application servers 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 November 1, 2010.

Copyright Notice

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 4
1.1. Motivation 4
1.2. Scope 5
1.3. Terminology 5
1.4. Discussion Venue 7
1.5. Contributors 7
1.6. Acknowledgements 7
2. Representation of Server Identity 8
2.1. Subject Naming in PKIX Certificates 8
2.2. PKIX Certificate Name Rules 9
3. Verification of Server Identity 10
3.1. Overview 10
3.2. Constructing an Ordered List of Reference Identifiers . . 10
3.3. Seeking a Match 12
3.4. Verifying a Domain Name 13
3.4.1. Checking of Traditional Domain Names 13
3.4.2. Checking of Internationalized Domain Names 13
3.4.3. Checking of Wildcard Labels 14
3.4.4. Checking of DNS Domain Names in Common Names 14
3.5. Verifying an Application Type 15
3.5.1. SRV-ID 15
3.5.2. URI-ID 15
3.6. Outcome 16
3.6.1. Case #1: Match Found 16
3.6.2. Case #2: No Match Found, Cached Certificate 16
3.6.3. Case #3: No Match Found, Uncached Certificate 16
3.6.3.1. Interactive User Agent 16
3.6.3.2. Automated Client 17
4. Security Considerations 17
4.1. Service Delegation 17
4.2. Wildcard Certificates 17
4.3. Internationalized Doman Names 18
5. IANA Considerations 18
6. References 18
6.1. Normative References 18
6.2. Informative References 19
Appendix A. Prior Art 22

A.1. IMAP, POP3, and ACAP (1999) 22
A.2. HTTP (2000) 23
A.3. LDAP (2000/2006) 25
A.4. SMTP (2002/2007) 28
A.5. XMPP (2004) 29

A.6. NNTP (2006) 30
A.7. NETCONF (2006/2009) 31
A.8. Syslog (2009) 32
A.9. SIP (2010) 33
A.10. GIST (2010) 34
Authors' Addresses 35

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 server 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 server using Transport Layer Security [TLS], it references some conception of the server's identity while attempting to establish a secure connection (e.g., "the web site at example.com"). Likewise, during TLS negotiation the server presents its conception of the server's identity in the form of a public-key certificate that was issued by a certification authority in the context of the Internet Public Key Infrastructure using X.509 [PKIX]. Only a match between the client's reference identity and the server's presented identity enables the client to be sure that the certificate can legitimately be used to secure the connection.

Many application technologies adhere to the pattern outlined here, 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 [XMPPBIS]
- 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]

- The Session Initiation Protocol [SIP], for which see also [SIP-CERTS]
- The General Internet Signalling Transport [GIST]

Application protocols have traditionally specified their own rules for representing and verifying server identities. Unfortunately, this divergence of approaches has caused some confusion among certification authorities, application developers, and protocol designers.

To codify best current practices regarding the implementation and deployment of secure PKIX-based authentication, this document specifies recommended procedures for representing and verifying server identities in certificates intended for use in applications employing TLS.

1.2. Scope

This document applies only to server identities associated with DNS domain names, only to TLS, and only to PKIX-based systems. Similar

considerations might apply to client identities (e.g., for mutual authentication), to identifiers other than DNS domain names (e.g., IP addresses), to security protocols other than TLS (e.g., [IPSEC] and [DTLS]), and to keys or certificates created outside the context of PKIX-based systems (e.g., on networks that use a web of trust model based on [OPENPGP] or 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)). Although those scenarios might be able to re-use some aspects of the representation and verification rules provided here, they are outside the scope of this document and need to be addressed by separate specifications.

1.3. Terminology

Because the concept of "identity" is too vague to be actionable in application protocols, we define a set of more concrete terms:

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

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

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 also used for matching purposes; the types covered in this specification are:

* CN-ID = a Relative Distinguished Name (RDN) of type Common Name (CN)

* DNS-ID = a subjectAltName identifier of type dNSName

* SRV-ID = the SRVName form of otherName from the GeneralName structure in SubjectAltName

* URI-ID = a subjectAltName identifier of type uniformResourceName

interactive client: A software agent or device that is directly controlled by a natural person.

PKIX certificate: An X.509v3 certificate generated and employed in the context of the Internet Public Key Infrastructure using X.509 [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 that is used by the client for matching purposes when checking the presented identifiers; the client can attempt to match multiple reference identifiers of different types.

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 URI scheme or a DNS SRV Service.

source domain: The DNS domain name that a client expects an application server to present in the certificate.

target domain: A domain name or host name that a client has derived from the soruce domain in an automated fashion (e.g., by means of a [DNS-SRV] lookup) or that a natural person directly controlling an interactive client has explicitly configured for connecting to the source domain.

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", "certificate", "credential", "identity", "self-signed certificate", "trust", "trust anchor", "trust chain", "validate", and "verify".

The following capitalized keywords are to be interpreted as described in [TERMS]: "MUST", "SHALL", "REQUIRED"; "MUST NOT", "SHALL NOT"; "SHOULD", "RECOMMENDED"; "SHOULD NOT", "NOT RECOMMENDED"; "MAY", "OPTIONAL".

1.4. Discussion Venue

[[RFC Editor: please remove this section.]]

The editors are actively seeking input from certification authorities, application developers, and protocol designers regarding the recommendations in this document. Please direct feedback to the editors directly or post to the <certid@ietf.org> mailing list, for which archives and subscription information are available at <https://www.ietf.org/mailman/listinfo/certid>.

1.5. Contributors

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

1.6. Acknowledgements

The editors and contributors wish to thank the following individuals for their feedback and suggestions: Nelson Bolyard, Scott Cantor, Dave Crocker, Cyrus Daboo, Philip Guenther, Bruno Harbulot, David Harrington, Paul Hoffman, Scott Lawrence, Alexey Melnikov, Ludwig Nussel, Joe Orton, Tom Petch, Tim Polk, Pete Resnick, Joe Salowey, and Dan Wing.

2. Representation of Server Identity

This section enumerates the rules for representing application server identity in PKIX certificates. First we provide a brief tutorial about subject naming, then we provide the rules.

2.1. Subject Naming in PKIX Certificates

The application server is the subject of a server certificate. As [PKIX] says, "[the] subject field identifies the entity associated with the public key stored in the subject public key field."

The application server is identified by a name or names carried in the subject field and/or the subjectAltName extension of the certificate. See sections 4.1.2.6 and 4.2.1.6 of [PKIX] for details. This section only briefly introduces distinguished names and their components, as well as subjectAltNames and the particular subjectAltName extension types explicitly mentioned in this specification.

The subject field of a PKIX Certificate is defined as a X.501 type Name [PKIX] and known as a Distinguished Name (DN). See also [X.501] A DN is an ordered sequence of attribute type and attribute value pairs (termed "attribute value assertions (AVAs)"), where the attributes are to be those of the subject. Each AVA is also termed a "Relative Distinguished Name (RDN)". The RDNs are ordered in the DN sequence from most general to most specific.

Note that certificates are binary objects -- they are encoded using distinguished encoding rules (DER). Thus, displayable (aka printable) renderings of certificate subject (and issuer) names means that the DER-encoded sequences are decoded and converted into a "string representation" of a DN before being rendered. Often such a DN string representation is ordered from left-to-right, most specific to most general. See [LDAP-AUTH] for details.

Certificate subjects may also have "alternative" names. These are represented within certificates using the SubjectAltName field. This field is a sequence of typed fields, where each type is a distinct type of identifier. For example, the SubjectAltName identifier types noted in this specification are:

- dNSName -- a DNS domain name [PKIX]
- SRVName -- a service name [SRVNAME]
- uniformResourceIdentifier -- a URI [URI] [PKIX]

2.2. PKIX Certificate Name Rules

The following rules apply to the representation of application server identities in PKIX certificates issued by certification authorities.

1. The certification authority MUST issue the certificate based on the DNS domain name (not an IP address) at which the server will provide the relevant service.

2. The certificate MUST include a "DNS-ID" (i.e., a subjectAltName identifier of type dNSName).

3. 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" (i.e., an instance of the SRVName form of otherName from the GeneralName structure in the subjectAltName as specified in [SRVNAME]).

4. 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 an URI-ID (i.e., a subjectAltName identifier of type uniformResourceIdentifier); the scheme SHALL be that of the protocol associated with the application type and the authority component SHALL be the DNS domain name of the service.

5. The certificate MAY include other application-specific identifiers for types that were defined before specification of the SRVName extension (e.g., XmppAddr for [XMPP]) or for which service names or URI schemes do not exist; however, such application-specific identifiers are not generally applicable and therefore are out of scope for this specification.

6. The DNS domain name portion of any identifier type MAY contain one instance of the wildcard character '*' (see [DNS-WILD]) but only as the left-most label of the DNS domain name component of the identifier (following the definition of "label" from [RFC1035]). An application protocol that employs the rules defined in this document MUST specify whether the wildcard character is or is not allowed in certificates issued for use with that protocol; by default wildcard certificates SHOULD be disallowed.

7. The certificate SHOULD NOT represent the server's DNS domain name in a Relative Distinguished Name (RDN) of type Common Name (CN)

(see [LDAP-SCHEMA]) within the subjectName field, even though it is recognized that many deployed clients still check for this

legacy identifier configuration within certificate subjectName. However, if this legacy identifer configuration is employed, then the server's DNS domain name MUST be placed in the last (most specific) RDN within the RDN sequence making up the certificate's subjectName, as the order of RDNs is determined by the DER-encoded Name within the server's PKIX certificate.

8. The certificate MUST NOT represent the server's DNS domain name by means of a series of Domain Component (DC) attributes (because the order of Domain Components is not guaranteed, certain attacks are possible if DC attributes are included).

3. Verification of Server Identity

3.1. Overview

At a high level, the client verifies the server's identity by performing the following actions:

1. Before connecting to the server, the client constructs an ordered list of reference identifiers against which to check the presented identifiers.

2. After the server provides its presented identifiers in the form of an PKIX certificate, the client checks each of its reference identifiers against the presented identifiers for the purpose of finding a match.

3. When checking a reference identifier against a presented identifier, the client (a) MUST match the source domain of the identifiers and (b) MAY also match the service type of the identifiers.

In addition to checking identifiers, a client MAY complete further checks to ensure that the server is authorized to provide the requested service. However, methods for doing so (e.g., consulting

local policy information) are out of scope for this document.

3.2. Constructing an Ordered List of Reference Identifiers

Before connecting to the server, the client MUST construct an ordered list of acceptable reference identifiers.

The inputs here 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 username of an IMAP or POP3 account for retrieving email), or some other combination of information that can yield a source domain and an application type.

The client might need to derive the source domain and application type from the input(s) it has received. The derived data MUST include only information that can be securely parsed out of the inputs (e.g., extracting the DNS domain name from the authority component of a URI or extracting the application type from the scheme of a URI) or information for which the derivation is performed in a secure manner (e.g., using [DNSSEC]).

In some cases the inputs might include more than one DNS domain name, because a user might have explicitly configured the client to associate a target domain with the source domain. Such delegation can occur by means of user-approved DNS SRV records (e.g., _xmpp-server._tcp.im.example.com might yield a hostname of hosting.example.net) or a user-configured lookup table for host-to-address or address-to-host translations (e.g., the Unix "hosts" file). See the Security Considerations for further discussion of service delegation.

Using the combination of DNS domain name(s) and application 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 DNS domain name that is (a) contained in or 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 target domain).

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

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

- The list SHOULD NOT include a CN-ID; however, the CN-ID (if included) MUST be constructed only from the source domain and never from a target domain.

  Note: A client MUST NOT construct a reference identifier corresponding to Relative Distinguished Names (RDNs) other than the Common Name and MUST NOT check for such RDNs in the presented identifiers. In particular, this means that a client MUST NOT check a series of Domain Component (DC) attributes if included in

  the certificate (because the order of Domain Components is not guaranteed, certain attacks are possible if DC attributes are checked).

The client then orders the list in accordance with the following rules:

- Reference identifiers that include the source domain MUST be preferred over reference identifiers that include a target domain (if any).

- Reference identifiers that include both a DNS domain name and an application type MUST be preferred over reference identifiers that include only a DNS domain name. Therefore an SRV-ID or URI-ID is to be preferred over a DNS-ID.

- A reference identifier of type CN-ID (if included) MUST always be the lowest-priority reference identifier in the list.

To illustrate the ordering rules, consider the case where the inputs are a source domain of "im.example.com" and an application type of "XMPP" (for which application servers are discovered via DNS SRV records) and the user of the client has also explicitly configured a target domain of "hosting.example.net". In this case, the ordered

list would be:

1. An SRV-ID of "_xmpp.im.example.com"
2. A DNS-ID of "im.example.com"
3. An SRV-ID of "_xmpp.hosting.example.net"
4. A DNS-ID of "hosting.example.net"
5. A CN-ID of "im.example.com"

3.3. Seeking a Match

Once the client has constructed its order list of reference identifiers and has received the server's presented identifiers in the form of an PKIX certificate, the client checks its reference identifiers against the presented identifiers for the purpose of finding a match. It does so by seeking a match in preference order and aborting the search if any presented identifier matches one of its reference identifiers. The search fails if the client exhausts its list of reference identifiers without finding a match. Detailed comparison rules for finding a match are provided in the following sections.

> Note: A client MUST NOT seek a match for a reference identifier of CN-ID if the presented identifiers include an SRV-ID, URI-ID, DNS-ID, or any application-specific subjectAltName extensions, and MUST NOT check a Common Name in the certificate if it is other than the last Relative Distinguished Name (RDN) in within the sequence of RDNs making up the Distinguished Name within the certificate's subjectName. (Note: The term "last" refers to the DER order, which is often not the string order presented to a user; the order that is applied here MUST be the DER order.)

3.4. Verifying a Domain Name

This document assumes that each reference identifier contains a DNS domain name that is a "traditional domain name" or an "internationalized domain name". The client MUST match the source domain of a reference identifier according to the following rules.

3.4.1. Checking of Traditional Domain Names

The term "traditional domain name" is a contraction of this more formal and accurate name: "traditional US-ASCII letter-digit-hyphen DNS domain name". Traditional domain names are defined in [RFC1034] and [RFC1035] in conjunction with [RFC1123] as further explained in [IDNA2003]. In essence, a traditional domain name consists of a set of one or more labels, with the labels usually shown separated by dots. Labels nominally consist of only [US-ASCII] uppercase and lowercase letters, digits, and hyphen. There are additional qualifications (please refer to the above-referenced specifications for details) but they are not germane to this specification.

If the source domain 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 components using a case-insensitive ASCII comparison (as clarified by [DNS-CASE]). Each label MUST match in order for the names to be considered to match.

3.4.2. Checking of Internationalized Domain Names

[[Editorial Note: This section needs to be updated to reflect [IDNA2008].]]

The term "internationalized domain name" refers to 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, as explained by [IDNA2003] and [IDNA2008].

If the source domain of a reference identifier is an internationalized domain name, then an implementation MUST convert the domain name to the ASCII Compatible Encoding (ACE) format as specified in Section 4 of [IDNA2003] before comparison; specifically, the conversion operation specified in Section 4 of [IDNA2003] MUST be performed 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.
- In step 5, change all label separators to U+002E (full stop).

After performing the "to-ASCII" conversion with regard to an internationalized domain name, the DNS labels and names MUST be compared for equality according to the rules specified in Section 3 of [IDNA2003], i.e. once all label separators are replaced with U+002E (dot) they are compared in a case-insensitive manner.

3.4.3. Checking of Wildcard Labels

Unless forbidden by an application protocol's own specification(s), an application protocol client employing this specification's rules MAY match the reference identifier against a presented identifier containing one instance of the wildcard character '*' (see [DNS-WILD]) as the left-most label of the domain name, e.g. *.example.com (following the definition of "label" from [RFC1035]).

If such a wildcard identifier is presented, the wildcard MUST be used to match only the one position-wise corresponding label (thus *.example.com matches foo.example.com but not bar.foo.example.com or example.com). The client MUST fail to match a presented identifier in which the wildcard character is contained within a label fragment (e.g., baz*.example.net is not allowed and MUST NOT be taken to match baz1.example.net and baz2.example.net), or in which the wildcard character does not comprise the left-most label in the presented identifier (e.g., neither bar.*.example.net nor bar.f*o.example.net are allowed).

3.4.4. Checking of DNS Domain Names in Common Names

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

Therefore, if and only if the identity set does not include subjectAltName extensions of type dNSName, SRVName, or uniformResourceIdentifier (or any application-specific subjectAltName extensions), the client MAY as a fallback check for a DNS domain name

in the value of the last Relative Distinguished Name (RDN), if it is of type Common Name (CN), within the sequence of RDNs making up the Distinguished Name within the certificate's subjectName. (Note: The term "last" refers to the DER order, which is often not the string

order presented to a user; the order that is applied here MUST be the DER order.)

In existing certificates, the CN is often used for representing a DNS domain name; for example, consider the following subjectName, where the last RDN is a Common Name that is intended to represent a DNS domain name:

ou=Web Services, c=GB, cn=www.example.com

Here the Common Name is "www.example.com" (a string whose form matches that of a DNS domain name) and the client could choose to compare a reference identifier of type CN-ID against that string. When doing so, the client MUST follow the comparison rules for the source domain of an identifier of type DNS-ID, SRV-ID, or URI-ID, as described under Section 3.4.

3.5. Verifying an Application Type

As specified under the ordering rules for reference identifiers, a client SHOULD check not only the domain name but also the application 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 a specific kind of service, such as a web site, an email service, or an instant messaging service.

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

3.5.1. 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.

3.5.2. 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.

3.6. Outcome

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

3.6.1. 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 server.

3.6.2. Case #2: No Match Found, Cached Certificate

If the client finds no presented identifier that matches any of the reference identifiers but a natural person has permanently accepted the certificate during a previous connection attempt, the certificate is cached. In this case, the client MUST verify that the presented certificate matches the cached certificate and MUST notify the user if the certificate has changed since the last time a secure connection was successfully negotiated.

3.6.3. Case #3: No Match Found, Uncached Certificate

If the client finds no presented identifier that matches any of the reference identifiers and a human user has not permanently accepted the certificate during a previous connection attempt, the client MUST NOT consider the certificate to include a validated identity for the application server.

Instead, the client MUST proceed as follows.

3.6.3.1. Interactive User Agent

If the client is an interactive client that is directly controlled by a natural person, then it MUST either do one of the following:

1. Automatically terminate the connection with a bad certificate error; or

2. Actively warn the user that the certificate is untrusted and encourage the user to terminate the connection, but give advanced users the option to (a) view the entire certificate chain, (b) accept the certificate either temporarily (i.e., for this connection attempt only) or permanently (i.e., for all future connection attempts), and then (c) continue with the connection attempt.

If a user permanently accepts a certificate, the client MUST cache

the certificate (or some non-forgeable representation such as a hash value) and in future connection attempts MUST behave as in "Case #2: No Match Found, Cached Certificate".

Note: It is the responsibility of the human user to verify the hash value or fingerprint of the certificate with the server over a trusted communication layer.

3.6.3.2. Automated Client

If the client is an automated application that is not directly controlled by a natural person, then it SHOULD terminate the connection with a bad certificate error and log the error to an appropriate audit log. An automated application MAY provide a configuration setting that disables this check, but MUST enable the check by default.

4. Security Considerations

4.1. Service Delegation

When the connecting application is an interactive client, the source domain name and application type MUST be provided by a human user (e.g. when specifying the server portion of the user's account name on the server or when explicitly configuring the client to connect to a particular host or URI as in [SIP-LOC]) and MUST NOT be derived from the user inputs in an automated fashion (e.g., a hostname address discovered through DNS resolution of the source domain). This rule is important because only a match between the user inputs (in the form of a reference identifier) and a presented identifier enables the client to be sure that the certificate can legitimately be used to secure the connection.

However, an interactive client MAY provide a configuration setting that enables a human user to explicitly specify a particular hostname (called a "target domain") to be checked for connection purposes.

4.2. Wildcard Certificates

Allowing 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].

4.3. Internationalized Doman Names

In addition to the wildcard certificate attacks previously mentioned, 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].

5. IANA Considerations

This document has no actions for the IANA.

6. References

6.1. Normative References

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

[IDNA2008] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", draft-ietf-idnabis-protocol-18 (work in progress), January 2010.

[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.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

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

6.2. 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.

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

[DNS-WILD]
Lewis, E., "The Role of Wildcards in the Domain Name System", RFC 4592, July 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.

[GIST] Schulzrinne, H. and M. Stiemerling, "GIST: General Internet Signalling Transport", draft-ietf-nsis-ntlp-20 (work in progress), June 2009.

[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.

[IDNA-DEFS] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", draft-ietf-idnabis-defs-13 (work in progress), January 2010.

[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.

[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.

[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.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC2459] Housley, R., Ford, W., Polk, T., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[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 B. Laboratories, "Domain Certificates in the Session Initiation Protocol (SIP)", draft-ietf-sip-domain-certs-06 (work in progress), March 2010.

[SIP-LOC] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[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.

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

[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.

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

[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.

[X.501] International Telecommunications Union, "Information Technology - Open Systems Interconnection - The Directory: Models", ITU-T Recommendation X.501, ISO Standard 9594-2, February 2001.

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

[XMPPBIS] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-xmpp-3920bis-07 (work in progress), April 2010.

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 server 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 server 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 server 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 [RFC2459]. 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 server 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 server 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 subjectName 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:

o in step 1, the domain name SHALL be considered a "stored string";

o in step 3, set the flag called "UseSTD3ASCIIRules";
o in step 4, process each label with the "ToASCII" operation; and
o 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 server 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 server 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] specified the following text regarding server 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 chain of certificates 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 certificate chain. 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, [XMPPBIS] refers to this document for rules regarding server identity verification in XMPP.

A.6. NNTP (2006)

In 2006, [NNTP-TLS] specified the following text regarding server 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 server 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 server 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 server 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.
- 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].

- 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)

At the time of this writing, [SIP-CERTS] specified text regarding server identity verification in the Session Initiation Protocol (SIP). However, that specification has not yet been approved by the IESG and text cannot be considered final.

The relevant text follows.

######

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. GIST (2010)

In 2010, [GIST] specified the following text regarding server 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 (editor)
Cisco

Email: psaintan@cisco.com

Jeff Hodges (editor)
PayPal

Email: Jeff.Hodges@PayPal.com