Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2010

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

March 8, 2010

# Representation and Verification of Application Server Identity in Certificates Used with Transport Layer Security (TLS)
## draft-saintandre-tls-server-id-check-03

## Abstract

Many application technologies enable a secure connection between two entities using certificates in the context of Transport Layer Security (TLS). This document specifies procedures for representing and verifying the identity of application servers in such interactions.

## Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 9, 2010.

## Copyright Notice

Table of Contents

# 1. Introduction

## 1.1. Motivation

When a client wishes to establish a secure communication channel with an application server (e.g., "the HTTP server for example.com"), at a minimum the client needs to verify the identity of the server in order to prevent certain passive and active attacks against the connection. To establish secure connections, many application technologies use Transport Layer Security [TLS] with certificates issued by certification authorities that are part of the Internet X.509 Public Key Infrastructure (PKI) as described in [X509]; such application technologies include but are not limited to the following:

- The Hypertext Transfer Protocol [HTTP], for which see also [HTTP-TLS]
- The Internet Message Access Protocol [IMAP] and the Post Office Protocol [POP3], for which see also [USINGTLS]
- The Lightweight Directory Access Protocol [LDAP], for which see also [LDAP-AUTH] and its predecessor [LDAP-TLS]
- The NETCONF Configuration Protocol [NETCONF], for which see also [NETCONF-SSH] and [NETCONF-TLS]
- The Network News Transfer Protocol [NNTP], for which see also [NNTP-TLS]
- The Session Initiation Protocol [SIP], for which see also [SIP-CERTS]
- The Simple Mail Transfer Protocol [SMTP], for which see also [SMTP-AUTH] and [SMTP-TLS]
- The Syslog Protocol [SYSLOG], for which see also [SYSLOG-TLS]
- The Extensible Messaging and Presence Protocol [XMPP], for which see also [XMPPBIS]

Different 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, protocol designers, and application developers.

Futhermore, currently the vast majority of deployed application servers use domain names in their certificates (typically via a subjectAltName extension of dNSName or a subjectName component of Common Name). Ideally, service operators would use application service identities in their certificates (such as an SRVName [SRVNAME], a URI, or an application-specific name form), since this would reduce the possibility of attacks against unrelated services at domain names that provide many different application services.





To codify secure authentication practices, this document specifies recommended procedures for representing and verifying server identities in certificates intended for use in applications that employ TLS.

## 1.2. Scope

This document applies only to server identities associated with domain names, only to TLS, and only to the PKI. Similar considerations might apply to client identities (e.g., for mutual authentication), to identities other than 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 the PKI (e.g., where a dependency on Certificate Revocation Lists or the Online Certificate Status Protocol might introduce unacceptable latency or denial of service attacks). 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

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", "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. Contributors

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

## 1.5. Acknowledgements

The editors and contributors wish to thank the following individuals for their feedback and suggestions: Dave Crocker, Cyrus Daboo, Philip Guenther, David Harrington, Paul Hoffman, Scott Lawrence, Alexey Melnikov, Tom Petch, Pete Resnick, Joe Salowey, and Dan Wing.





# 2. Architectural Assumptions

Internet applications often use a client-server architecture in which a client connects to a server in order to retrieve or upload data, access a broader network of services, or communicate with other entities on the network. From the security perspective, a client might be a user agent controlled by a human, an automated process such as a bot, or a peer server. We assume that an application server hosts information, enables a provisioned account to perform authorized services, or provides network access on behalf of a particular organization or service that is canonically identified by a domain name (not, e.g., an IP address). The specific application provided (e.g., a web site or an instant messaging system) might further restrict the identity type that is represented or verified in a TLS interaction; such an identity type is often informal (e.g., most people expect a service that is provided on port 443 to be a web server) but can be specified more formally through the use of DNS SRV records [DNS-SRV], a Uniform Resource Identifier [URI] represented by a subjectAltName of uniformResourceIdentifier, the SRVName certificate extension defined in [SRVNAME], or other certificate extensions that have been defined for particular identity types (e.g., the XmppAddr extension for use in [XMPP]). The certificate

presented by an application server might contain one identity or it might contain multiple identities of different types (e.g., one identity might be a simple dNSName and another might be an SRVName that restricts the certificate to use in the context of the specified service).

When a client connects to an application server, it has some conception of either the server's identity (e.g., "an XMPP server for example.com") or of the server's location (e.g., "the SMTP server at mail.example.com"). The client expects at least one of the server's presented identities to match this reference identity. It is important to note that the reference identity is provided by a human user or configured into the client application (e.g., in the domain part of an "Application Unique String" as described in [SIP-LOC]), and is not an identity that is derived from the reference identity in an automated fashion (e.g., an IP address discovered through DNS resolution of the reference identity). Only a match between the reference identity and a presented identity enables the client to be sure that the certificate can legitimately be used to secure the connection. However, a user-oriented client MAY provide a configuration setting that enables a human user to explicitly specify a particular hostname to be checked for connection purposes, thus explicitly overriding matching rules.

To summarize, we define the following terms to assist in understanding the process of representing and verifying server





identity:

identity set: The set of identities that are presented by a server to a client (in the form of the server's X.509 certificate) when the client attempts to establish a secure connection with the server.

identity type: The "natural kind" of identity to which a presented identity or reference identity belongs. For example, the reference identity might be a domain name, an IPv4 or IPv6 address, or a particular service in the sense of [DNS-SRV] or [SRVNAME]. This specification does not provide a complete taxonomy of identity types but assumes that an application server is identified by a domain name that could be represented in the form of several identity types (e.g., a dNSName and an SRVName).

presented identity: A single member of the identity set.

reference identity: The client's conception of the server's identity before it attempts to establish a secure connection with the server, i.e. the identity that the client expects the server to present and to which the client makes reference when attempting to verify the server's identity. The reference identity MUST be provided by the human user controlling the client (if any), 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. The reference identity MAY be reflected in the TLS "server_name" extension as specified in [TLS].

## 3. Representation of Server Identity

The following rules apply to the representation of application server identities in certificates issued by certification authorities, i.e., certificates that are to used for securing connections to application servers such as web sites, email services, and instant messaging services.

1. The certification authority MUST issue the certificate based on the domain name at which the server will provide the relevant service (not an IP address or host name for a specific machine).
2. An identity MAY contain one instance of the wildcard character '*' but only as the left-most label. An application protocol that re-uses the rules specified in this document MUST specify whether the wildcard character is or is not allowed in certificates issued for use with that protocol.
3. If the service at which the certificate will be used deploys a technology that is discovered by means of DNS SRV records [DNS-SRV], then the certificate SHOULD include an identity of type SRVName [SRVNAME].





4. If the service is associated with a particular URI, then the certificate MAY include an identity of type uniformResourceIdentifier (i.e., a subjectAltName extension that includes a uniformResourceIdentifier name form whose authority component is the domain name of the service).
5. Although the certificate MAY include other application-specific identities for types that were defined before specification of the SRVName extension (e.g., XmppAddr for [XMPP]) or for which

service names do not exist, the SRVName extension is preferred.

6. If the certificate does not include an SRVName, uniformResourceIdentifier, or other application-specific identity type, then the certificate MUST include an identity of dNSName.
7. The certificate SHOULD NOT represent the server's domain name in an identity of Common Name (CN) (see [LDAP-SCHEMA]) in the leaf Relative Distinguished Name (RDN) of the subjectName, even though it is recognized that many deployed clients still check this legacy identity. For example, here the CN is the leaf RDN, which is acceptable:

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

8. The certificate MUST NOT represent the server's domain name in an identity of Common Name (CN) where the CN is in something other than the "leaf" (left-most) position within the Relative Distinguished Names (RDNs) of the subjectName. For example, here the CN is not the leaf RDN, which is unacceptable:

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

9. The certificate MUST NOT represent the server's 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).

# 4. Verification of Server Identity

## 4.1. Overview

At a high level, the client verifies the server's identity in accordance with the following rules:

1. Before connecting to the server, the client determines the possible identity type(s) of the reference identity.
2. During the process of attempting to establish a secure connection, the server MUST present its identity set to the client in the form of an X.509 certificate [X509].





3. Upon being presented with the server's identity set, the client

MUST check the reference identity against the presented identities for the purpose of finding a match. To do so, the client iterates through all of the subjectAltName extensions it recognizes in the server's certificate and compares the value of each extension to the reference identity until it has either produced a match or exhausted the identities in the identity set (comparison rules for matching particular identity types are provided under Section 4.2, including fallbacks to several subjectName fields). Even if the application technology does not define an application-specific preference order for checking of identities, the client SHOULD check them in order from most specific (e.g., SRVName) to least specific (e.g., dNSName).

4. Before attempting to find a match in relation to a particular presented identity, the client MAY map the reference identity to a different identity type. Such a mapping MAY be performed for any available subjectAltName type to which the reference identity can be mapped; however, the reference identity SHOULD be mapped only to types for which the mapping is inherently secure (e.g., extracting the domain name from a URI to match against a subjectAltName of type dNSName or SRVName).
5. The client MUST NOT attempt to match against an an identity of type SRVName [SRVNAME] unless the service to which the client has connected deploys a technology that is discovered by means of DNS SRV records [DNS-SRV].
6. If the identity set has more than one member, a match with any of the presented identities is acceptable.

Note: A hostname that is resolved via the Domain Name System MUST NOT be used as the reference identity unless an administrator or human user has explicitly configured an application to associate a particular hostname (and potentially port) with the hostname to which the application connects (e.g., to "hardcode" an association between an original hostname of example.net and a configured hostname of connector.example.com:443).

In addition to the identity check described in this section, a client might complete further verification to ensure that the server is authorized to provide the service it is requested to provide. Methods for doing so (which might include consulting local policy information) are out of scope for this document.

## 4.2. Comparison Rules

This document assumes that the reference identity is a domain name as defined by [RFC1034] and [RFC1035] for "traditional" domain names or by [IDNA2003] or [IDNA2008] for internationalized domain names. The client MUST match the reference identity against subjectAltName

extensions of type dNSName and SRVName [SRVNAME] according to the following rules.

### 4.2.1. Traditional Domain Names

If the reference identity is a "traditional" domain name, then matching of reference identity against the presented identity is performed by comparing the set of domain components using a case-insensitive ASCII comparison (as clarified by [DNS-CASE]).

### 4.2.2. Internationalized Domain Names

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

If the reference identity is an internationalized domain name, then an implementation MUST convert the reference identity to the ASCII Compatible Encoding (ACE) format as specified in Section 4 of [IDNA2003] before comparison with subjectAltName values of type dNSName; 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.

### 4.2.3. Wildcards

Unless otherwise specified by an application protocol that re-uses the rules specified in this document, the client MAY match against an identity that contains one instance of the wildcard character '*' as the left-most label of the domain name. If it does so, the client MUST match the reference identity against the entire left-most label only (thus *.example.com matches foo.example.com but not bar.foo.example.com or example.com itself). The client MUST ignore a presented identity 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).

#### 4.2.4. Common Names

If and only if the identity set does not include subjectAltName extensions of type dNSName, SRVName, uniformResourceIdentifier (or other application-specific subjectAltName extensions), the client MAY as a fallback check the value of the Common Name (CN) if presented as the leaf (left-most) Relative Distinguished Name (RND) in the subjectName component of the server's X.509 certificate. In existing certificates, the CN is often used for representing a domain name; for example, consider the following subjectName with a leaf RDN of Common Name:

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

Here the Common Name is "www.example.com" and the client could choose to compare the reference identity against that CN. When doing so, the client MUST follow the comparison rules described above for subjectAltName extensions of type dNSName and SRVName.

A client MUST NOT check the Common Name if it is other than the leaf (left-most) Relative Distinguished Name (RDN) in the subjectName.

#### 4.2.5. Relative Distinguished Names

A client MUST NOT check Relative Distinguished Names (RDNs) other than the Common Name; in particular, this means that a series of Domain Component (DC) attributes MUST NOT be checked (because the order of Domain Components is not guaranteed, certain attacks are possible if DC attributes are checked).

### 4.3. Outcome

The outcome of the checking procedure is one of the following:

Case #1: The client finds at least one presented identity that matches the reference identity; the entity MUST use this as the validated identity of the server.

Case #2: The client finds no subjectAltName or subjectName that matches the reference identity but a human user has permanently

accepted the certificate during a previous connection attempt; the client MUST verify that the cached certificate was presented and MUST notify the user if the certificate has changed since the last time that a secure connection was successfully negotiated.

Case #3: The client finds no subjectAltName or subjectName that matches the reference identity and a human user has not permanently accepted the certificate during a previous connection attempt. The client MUST NOT use the presented identity (if any). Instead, if the client is a user-oriented application, then it





MUST either (1) automatically terminate the connection with a bad certificate error or (2) show the certificate (including the entire certificate chain) to the user and give the user the choice of terminating the connecting or accepting the certificate temporarily (i.e., for this connection attempt only) or permanently (i.e., for all future connection attempts) and then continuing with the connection; if a user permanently accepts a certificate in this way, the client MUST cache the certificate (or some non-forgeable representation such as a hash value) and in future connection MUST attempt behave as in Case #2. (It is the resposibility of the human user to verify the hash value or fingerprint of the certificate with the server over a trusted communication layer.) If the client is an automated application, 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.

## 5. Security Considerations

This entire document discusses security.

## 6. IANA Considerations

This document has no actions for the IANA.

## 7. References

### 7.1. 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.

[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.

[X509] 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.

## 7.2. Informative References

[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.

[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.

[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.

[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.

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

[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-04 (work in progress), May 2009.

[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.

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

[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-04 (work

in progress), November 2009.

# 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.

- o 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:

o 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 leaf Relative Distinguished Name (RDN) of the subjectName field of the server's certificate. 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 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:





- 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 [X509]).
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 [X509]. 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 [X509] 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 [X509] 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 [X509]. 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 [X509].
- 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 [X509], 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 [X509].

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". [X509], 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.

######

Authors' Addresses

Peter Saint-Andre (editor)
Cisco

Email: psaintan@cisco.com

Jeff Hodges (editor)
PayPal

Email: Jeff.Hodges@PayPal.com