None
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2010

P. Saint-Andre
Cisco
K. Zeilenga
Isode Limited
J. Hodges
PayPal
R. Morgan
Internet2

October 2, 2009

Server Identity Verification in Application Protocols
draft-saintandre-tls-server-id-check-02

Status of this Memo

Copyright Notice

Abstract

Technologies such as Transport Layer Security (TLS) and IPsec enable a secure connection between two entities (a "client" and a "server") using X.509 certificates. This document specifies recommended procedures for checking the identity of the server in such an interaction.

Table of Contents

1. Introduction

Technologies such as Transport Layer Security [TLS] and [IPSEC] enable a secure connection between two entities using the Internet X.509 Public Key Infrastructure (PKI) as described in [X509]. In such interactions, the entity that initiates the connection is called a "client" and the entity that receives the connection is called a "server".

> Note: The terms "client" and "server" as used here refer to security roles, not application roles; a server in the context of TLS or IPSec might be a "client" (i.e., a user agent) in the context of an application protocol as deployed on the Internet.

If a client wishes to connect to a server securely, it needs to check the identity of the server so that it can determine if the server is what it claims to be, verify that there is no attacker in the middle, and enforce other relevant security considerations. Typically this checking is done by correlating the information presented in the server's certificate with information available about the server contained in the Domain Name System (DNS) or provided by a human user.

Different application protocols that make use of the client-server pattern for security purposes have traditionally specified their own procedures for checking server identities. Examples include but are not limited to:

- 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]

Unfortunately, this divergence of approaches has caused some confusion among developers and protocol designers. Therefore this

document specifies recommended identity checking procedures for application protocols produced within the Internet Standards Process, for the purpose of codifying secure authentication practices.

Note: This document is currently limited in scope to the presentation of identities in X.509 certificates as issued in the context of the Public Key Infrastructure (PKI) and as applied to Transport Layer Security [TLS]; a future version of this document might address X.509 certificates as issued outside the context of the PKI, non-X.509 public keys such as OpenPGP keys, presentation of identities in ways other than in the certificate itself (e.g., certificate fingerprints for Secure Shell as described in [SSH] or for Datagram Transport Layer Security DTLS and Secure Real-time Transport Protocol as described in [DTLS-SRTP]), and applications that use security technologies other than TLS.

2. Conventions

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".

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

In addition, we define the following terms to assist in understanding the process of verifying server identity:

identity set: The set of identities that are presented by the server to the client (in the form of the server's X.509 certificate) when the client attempts to establish a secure connection to 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, an email address, a SIP address, an XMPP address, or some other type (this specification does not yet provide a complete taxonomy of identity types). In the case of domain names, the reference identity MUST NOT contain the wildcard character '*' (ASCII 42) in the left-most (least significant) domain name component or component fragment.

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 to 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. It is either the address to which the client connected or the explicit value of the TLS "server_name" extension as specified in [TLS]. The reference identity might be based on a DNS lookup, user configuration, or some other mechanism.

3. Verification Process

When a client connects to a server, it MUST verify the server's identity in order to prevent certain passive and active attacks against the connection. By "verify identity" we mean that the client needs to establish that at least one of the presented identities matches the reference identity.

3.1. Overview

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

1. Before connecting to the server, the client determines the identity type 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 (potentially in an application-specific preference order) and compares the value of each extension against 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 3.2, including fallbacks to several subjectName fields).
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 either inherently secure (e.g., extracting the DNS name from a URI to compare with a

subjectAltName of type dNSName or SRVName) or for which the mapping is performed in a secure manner (e.g., using [DNSSEC], or using a user-configured or admin-configured lookup table for host-to-address or address-to-host translations).
5. If the identity set has more than one member, a match with any of the presented identities is acceptable.

Note: Beyond the server identity check described in this section, clients might complete further checking to ensure that the server is authorized to provide the service it is requested to provide. The client might need to make use of local policy information in making this determination.

3.2. Comparison Rules

3.2.1. Domain Names

If the reference identity is a domain name as defined by [RFC1034] and [RFC1035] for "traditional" domain names or by [IDNA] for internationalized domain names, then the client can match the reference identity against subjectAltName extensions of type dNSName and SRVName [SRVNAME] according to the following rules.

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.

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 [IDNA] before comparison with subjectAltName values of type dNSName; specifically, the conversion operation specified in Section 4 of [IDNA] 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 [IDNA].

Unless otherwise specified by an application protocol, the dNSName MAY contain one instance of the wildcard character '*'. The wildcard character applies only to the left-most domain name component and

matches any single component (thus a dNSName of *.example.com matches foo.example.com but not bar.foo.example.com or example.com itself). The wildcard character is not allowed in component fragments (thus a dNSName of baz*.example.net is not allowed and shall not be taken to match baz1.example.net and baz2.example.net).

In addition to checking the subjectAltName extensions of type dNSName and SRVNAME, the client MAY as a fallback check the value of the

Common Name (CN) (see [LDAP-SCHEMA]) as presented in the subjectName component of the server's X.509 certificate. In existing certificates, the CN is often used for encapsulating a domain name; for example, consider the following subjectName:

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 comparing the referenced identity against the Common Name, the client MUST follow the comparison rules described above for subjectAltName extensions of type dNSName and SRVName, with the exception that no wildcard matching is allowed.

In order to match domain 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).

3.2.2. IP Addresses

If the reference identity is an IP address as defined by [IP] or [IPv6], then the client can match the reference identity against subjectAltName extensions of type iPaddress according to the following rules.

The reference identity MUST be converted to the "network byte order" octet string representation; for IP Version 4 the octet string will contain exactly four octets, and for IP Version 6 the octet string will contain exactly sixteen octets. The client then compares this octet string, where a match occurs if the reference identity and presented identity octet strings are identical.

3.2.3. Email Addresses

If the reference identity is an email address as defined by [EMAIL], then the client SHOULD compare the reference identity against the value of the "rfc822Name" subjectAltName extension described in

[X509].

The client MAY also compare the reference identity against the value of the "E" attribute of the subjectName as described in [CRMF].

3.2.4. SIP Addresses

If the reference identity is a SIP address as defined by [SIP], then the client SHOULD compare map the reference identity to a domain name or email address and proceed as described for those identity types, or proceed as described in [SIP-CERTS].

3.2.5. XMPP Addresses

If the reference identity is an XMPP address ("JabberID") as defined by [XMPP], then the client SHOULD compare the reference identity against the value of the following subjectAltName extensions, in this order: SRVName, dNSName, and (as defined in [XMPP]) "id-on-xmppAddr".

3.3. 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 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 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) as the validated identity of the server and instead MUST proceed as described in the next section. 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 attempts 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 peer 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 provide a setting that enables the check.

4. Security Considerations

This entire document discusses security.

5. IANA Considerations

This document has no actions for the IANA.

6. References

6.1. Normative References

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

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

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

[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.

6.2. Informative References

[CRMF] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211,

September 2005.

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

[DTLS-SRTP] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)", draft-ietf-avt-dtls-srtp-07 (work in progress), February 2009.

[EMAIL] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[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.

[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.

[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.

[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.

[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.

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

[SSH] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[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.

[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-02 (work in progress), September 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:

o 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 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 [IDNA] 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.

######

As of this writing, [XMPPBIS] specified updated text regarding server identity verification in XMPP. However, that specification has not yet been approved by the IESG, and the relevant text might be

replaced by a reference to this document.

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 (2009)

As of this writing, [SIP-CERTS] specified text regarding server identity verification in SIP. However, that specification has not yet been approved by the IESG, and the relevant text might be replaced by a reference to this document.

Authors' Addresses

Peter Saint-Andre
Cisco

Email: Peter.SaintAndre@WebEx.com

Kurt D. Zeilenga
Isode Limited

Email: Kurt.Zeilenga@Isode.COM

Jeff Hodges
PayPal

Email: Jeff.Hodges@KingsMountain.com

RL 'Bob' Morgan
UWashington/Internet2

Email: rlmorgan@washington.edu