None
Internet-Draft
Intended status: Standards Track
Expires: March 4, 2010

P. Saint-Andre
Cisco
K. Zeilenga
Isode Limited
J. Hodges
PayPal
R. Morgan
Internet2
August 31, 2009

Server Identity Verification in Application Protocols
draft-saintandre-tls-server-id-check-01

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, etc. Typically this 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).

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 predecesor [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 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, "assurance", "attack", "authentication", "authorization", "certificate", "certification authority", "confidentiality", "credential", "downgrade", "encryption", "fingerprint", "hash value", "identity", "integrity", "signature", "security perimeter", "self-signed certificate", "sign", "spoof", "tamper", "trust", "trust anchor", "trust chain", "validate", "verify".

In addition, we define the following terms to assist in understanding the process of verifying 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 is 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, a JabberID, 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; this is 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.

3. Verification Process

When a client connects to a server, it MUST verify the server's identity (in order to prevent passive and active attacks against the connection). By "verify identity" we mean that the client needs to establish that at least one of the identities in the identity set 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 for which the mapping is performed in a secure manner (e.g., using DNSSEC, or using user-configured or admin-configured host-to-address/address-to-host lookup tables).

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, the DNS labels and names MUST be compared for equality according to the rules specified in Section 3 of [IDNA].

A dNSName MAY contain the wildcard character '*' (ASCII 42). The wildcard character applies only to the left-most (least significant) domain name component or component fragment and matches any single component or component fragment. For instance, a dNSName of *.example.com matches foo.example.com but not bar.foo.example.com or example.com itself; similarly, a dNSName of baz*.example.net matches baz1.example.net and baz2.example.net but not qux.example.net or example.net itself.

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. JabberIDs

If the reference identity is a JabberID as defined by [XMPP], then the client SHOULD compare the reference identity against the value of the "id-on-xmppAddr" subjectAltName extension of type otherName described in [XMPP], or proceed as described in [XMPPBIS].

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

To follow.

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.

[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.

[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-01 (work in progress), August 2009.

Authors' Addresses

Peter Saint-Andre
Cisco

Email: psaintan@cisco.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