The Use of Transport Layer Security (TLS) in the Session Initiation Protocol (SIP)
draft-gurbani-sip-tls-use-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 August 30, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document explores the use of the Transport Layer Security (TLS) in the Session Initiation Protocol (SIP). We do so by outlining the goals and the non-goals for the use of TLS and SIP. In doing so, a number of open questions are encountered regarding the contents of certificates and the behavior of SIP entities using such certificates. We hope to foster discussion in the SIP working group.
on these issues.

Table of Contents

1. Terminology .............................................. 3
2. Introduction .......................................... 3
3. Goals and non-goals of TLS use in SIP .................... 4
4. Security Analysis ....................................... 5
5. Open questions ......................................... 6
   5.1 An Authoritative Proxy ............................... 6
   5.2 Mutual Authentication ............................... 6
   5.3 URI Promotion ....................................... 6
   5.4 TLS Site Certificates and RFC3263 ................... 7
   5.5 Leveraging the Via Trail ............................. 8
6. Security Considerations ................................ 9
7. Acknowledgments ........................................ 9
8. References ............................................... 9
   8.1 Normative References ................................ 9
   8.2 Informative References .............................. 10
   Authors’ Addresses .................................... 10
A. TLS in SIP Test Cases .................................. 10
   A.1 Tests for valid certificates ....................... 11
      A.1.1 Good certificate, Case 1 ....................... 11
      A.1.2 Good certificate, Case 2 ....................... 11
   A.2 Tests for invalid certificates ................... 11
      A.2.1 Invalid certificate, Case 1 .................. 11
      A.2.2 Invalid certificate, Case 2 .................. 11
      A.2.3 Invalid certificate, Case 3 .................. 12
      A.2.4 Invalid certificate, Case 4 .................. 12
      A.2.5 Invalid certificate, Case 5 .................. 12
   A.3 Certificate depth test ............................... 12

Intellectual Property and Copyright Statements ............ 13
1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

2. Introduction

TLS [3] has started to appear in increasing number of SIP implementations. In order to aid in the interoperability and the uniform implementation of TLS in SIP, this document explores the use of TLS in SIP. In doing so, a number of open questions are encountered regarding the contents of certificates and the behavior of SIP entities using such certificates. This document catalogues these issues to open a discussion on them in the SIP Working Group.

In addition, based on our experience with implementing TLS in a SIP stack, we have derived a list of test cases. These are documented in Appendix A. These test cases may be of interest to the SIP Interoperability team and to developers who are currently adding support TLS in SIP.

For the discussion that follows, we assume the standard SIP trapezoid shown in Figure 1:

```
P1 -------------------------------------- P2
(proxy.example.com)                   (proxy.example.net)
V                                       V
/                                        /
/                                        /
/                                        /
/                                        /
User Agent                              User Agent
(sip:alice@example.com)                 (sip:bob@example.net)
```

Figure 1: Traditional SIP Trapezoid.

alice@example.com registers with her proxy (sip:proxy.example.com). Unless otherwise stated, we will assume that end points do not posses certificates, and that proxies, registrars and redirect servers do. Thus, Alice uses a sip scheme in her registration (as opposed to a sips scheme). Likewise, Bob registers a contact using a sip scheme with his proxy (sip:proxy.example.net). Both P1 and P2 posses X.509 certificates and support TLS.

Alice sends a request to Bob through her proxy. Based on either the presence of a pre-loaded sips URI in the Route header corresponding
to Bob’s proxy, or because of a DNS NAPTR lookup that resulted in the choice of TLS as the transport, Alice’s proxy (P1) opens up a TLS session with Bob’s proxy (P2). Thus communications between P1 and P2 are deemed secure.

In this document, we are concerned solely with the security of SIP signaling traffic. For a complete solution, media traffic must be secured as well; however, that is out of scope of the current discussion.

3. Goals and non-goals of TLS use in SIP

A detailed analysis of a threat model in SIP is also available in [2]. The threat model deals with the following possible attacks:

1. Outsider attack: An attacker mounts an attack without any certificates.
2. Insider attack: An attacker mounts an attack with a valid certificate for a domain.
3. Eavesdropping attack: All messages between two entities are routed correctly, but may be read by the attacker.
4. Man-in-the-Middle (MiTM) attack: Messages may be modified by the attacker.

The goals, then, of using TLS in SIP are such that security is assumed across the following dimensions:

1. Confidentiality: SIP message confidentiality should be protected from outsider MiTM attack, and from insider eavesdropping attack.
2. Integrity: SIP message integrity should be protected from MiTM attack.
3. Authenticity: Mutual authentication should occur between two proxies communicating over TLS; i.e., with respect to Figure 1, P1 must rely that it has indeed established connection with P2, and vice-versa.
4. Non-repudiation: Mutual non-repudiation of proxies that are part of a TLS connection must be assured; i.e., P1 cannot later plausibly deny contacting P2.

The specific non-goals of using TLS in SIP are:

1. Confidentiality and integrity in the presence of insider MiTM attack is not ensured.
2. Authenticity and non-repudiation at the SIP application layer is not ensured.
3. Access controls, privacy, availability or communication security are explicit non-goals when using TLS.
4. Security Analysis

The use of TLS to achieve the goals outlined above is standard. It requires that for proxy.example.net to be trusted by proxy.example.com, proxy.example.net must have been issued a certificate containing the canonical host name as Distinguished Name (DN) of the Subject field, or a DNS field entry in subjectAltName X.509v3 extension. Furthermore, the certificate for proxy.example.net must be backed by a certificate chain with a trust anchor trusted by proxy.example.com as well. When proxy.example.com checks proxy.example.net’s certificate for validity, besides the normal checks (certificate date validation, and if possible, revocation) it also ensures that proxy.example.net’s trust anchor is trusted by proxy.example.com and that proxy.example.net is the DN of the Subject field or one of the URIs in the subjectAltName X.509v3 extension.

Symmetric conditions are required for proxy.example.net to trust proxy.example.com. Once the trust chain is established, the goals -- confidentiality, integrity, authenticity, and non-repudiation -- are met by the use of TLS in SIP.

However, establishing a TLS trust chain does nothing to mitigate the non-goals. Confidentiality and integrity can be violated by an insider MiTM attack. Consider, for instance, an attacker with a valid certificate for example.com that poisons DNS such that requests for example.net end up at the attacker’s node in the example.com domain. Furthermore, if an attacker in example.com somehow gains access to the private key of an unsuspecting victim, then the attacker can masquerade as the victim for at least the length of time it takes the victim to find out that his key has been compromised.

Establishing a TLS link also does nothing to mitigate authenticity and non-repudiation at the SIP application layer. A TLS link authenticates both the end points at each end of of the link, however, it does not authenticate or provide non-repudiation of discrete SIP messages flowing through the link. Consider for example the case that a TLS link between two proxies may carry traffic for more than one transaction (or dialog) between the proxies. Thus a malicious host in one domain may well inject suspect SIP traffic in the other domain, and this sort of attack cannot be detected or prevented by the endpoints at either end of the TLS link. When endpoints use TLS, there aren’t any checks at the SIP layer correlating the contents of the TLS certificate with the SIP headers. In the presence of normal redirection, a receiving TLS entity cannot even enforce that the domain of the URI in the From header correspond to that of the sender’s domain as asserted by the sender’s certificate.
Access control, privacy, availability and communication security are out of scope of TLS.

5. Open questions

In this section, we catalogue some open questions on the use of TLS in SIP.

5.1 An Authoritative Proxy

When TLS is used, RFC3261 [2] instructs that proxy servers, registrars, and redirect server should possess a site certificate whose subject corresponds to the canonical name of the host. However, no provision is made to ensure that the host is indeed authoritatively authorized to act as a proxy for that domain. Is the dissemination of this trait considered of interest to the WG? If so, are there provisions in X.509 that allow the conveyance of this information? Or would an alternate mechanism like Attribute Certificates [4] or SAML be more appropriate here?

5.2 Mutual Authentication

With reference to Figure 1, when P1 establishes a TLS link with P2, mutual authentication should occur. P1 should ensure that P2’s certificate contains P2’s canonical hostname. At P1, matching the canonical hostname in the presented certificate to the intended destination is trivial since P1 obtained the intended destination possibly through a DNS query following the procedures in [5].

However, P2 accepted the connection as a passive listener. Thus, it cannot depend on accepting, in a blind fashion, the contents of P1’s certificate that contains P1’s canonical hostname. For all P2 knows, P1 may have usurped someone else’s legitimate certificate and is now trying to establish a connection and present a stolen certificate. Thus, to be certain, P2 should do a reverse DNS lookup of the source address of the incoming TCP packet and match the result to the contents of the certificate that contain P1’s canonical hostname.

Even this is not entirely foolproof since P1 could have forged the source IP address to match the contents of the certificate.

The rules for mutual authentication should be better specified in the standard. Is what is detailed above enough? Or do we need more?

5.3 URI Promotion

Should a sip URI be promoted to a sips URI based on the NAPTR/SRV lookups that resulted in the choice of TLS as the preferred
transport? If this is not done, it is possible for a request received over TLS at the recipient to be sent out over a non-TLS link. As an example, consider the trapezoid of Figure 1. Assume that Alice sends a request to "sip:bob@example.net". P1 gets the request and runs RFC3263 server resolution on it to derive a TLS as a preferred transport. P1 sends the request to P2, albeit with a R-URI of "sip:bob@example.net". Bob has his forwarding on so that all incoming requests are forwarded to "sip:bob@example.org". When P2 attempts to contact Bob, it will do so by proxying the request over a possibly non-secure link.

Ostensibly, if Bob was paranoid, he could have registered a sips URI as his forwarding address. Then the right thing would happen. Also, example.org domain may have DNS resolution set up in a manner such that TLS is the preferred transport. However, the security in this case depends on either Bob to do the right thing, or the domain to give priority to TLS DNS registrations. On the other hand, if from the onset a sip URI is promoted to a sips URI, the security of the signaling is made much more explicit.

5.4 TLS Site Certificates and RFC3263

Section 4.1 in [5] states that,

For NAPTR records with SIPS protocol fields, (if the server is using a site certificate), the domain name in the query and the domain name in the replacement field MUST both be valid based on the site certificate handed out by the server in the TLS exchange. Similarly, the domain name in the SRV query and the domain name in the target in the SRV record MUST both be valid based on the same site certificate.

There has been a question raised on the SIP implementors mailing list (see https://lists.cs.columbia.edu/pipermail/sip-implementors/2005-October/010547.html) as to what is the intent of this section? Is the intent that the NAPTR replacement field values be part of the certificate as well? Or that if there are multiple SRV RRs, then all of these should be part of the certificate? Or is the intent that a single high-level domain name (i.e., example.com) be in the site certificate instead of discrete servers in that domain (i.e., server1.example.com, server2.example.com, and so on).

This brings up a related question: what constitutes a site certificate? RFC3261 states that, "Proxy servers, redirect servers, and registrars SHOULD possess a site certificate whose subject corresponds to their canonical hostname." However, what does this mean when a SRV lookup on example.com returns multiple SIP servers: does each server have a unique certificate with its canonical
hostname embedded in the certificate? Or does each server have a high level FQDN (i.e., example.com) in the certificate and the verifier must somehow trust that the server they are establishing a connection with (server1.example.com) is indeed aptly represented by a certificate whose DN (or subjectAltName) contains "example.com"?

In the latter case, it seems to be assumed that the site certificate is shared among the many servers. This implies that the private key is shared as well, which means that a compromise of the private key at any one of the hosts would render the entire site to be insecure.

Current drafts seem to support the notion that a collection of hosts that make up a SRV RR set share a certificate; certainly, [6] assumes this behavior.

5.5 Leveraging the Via Trail

In Section 4, we discussed how the authentication and non-repudiation non-goals mitigate the use of TLS. Unlike HTTPS [9], which uses TLS to secure the transport layer and translate this directly to application layer security (e.g., the "padlock" icon in most browsers), the use of SIPS does not quite do the same. HTTPS has a simple end-to-end model compared to SIP’s hop-by-hop proxy model, thus in HTTP, when the padlock is indeed locked, the user can be assured that the traffic is flowing in a secure fashion from the browser to the server. In SIP, by contrast, the use of TLS can be assured only between hops. A proxy in a chain far removed from another proxy much later in the chain cannot authoritatively determine whether the later proxy is indeed using TLS. All it can state with any certainty is that its neighbors have used TLS. Furthermore, the use of TLS itself has no bearing on the traffic flowing through the TLS link. Suspect SIP messages may well be carried over this link, and as long as they are well-formed, the two endpoints at each end of the TLS link do not, and typically cannot enforce any rules that prohibit the passage of these messages. Thus it is quite possible for a cracker to compromise a host in one domain, and use the TLS link between that domain and a peer domain to transfer suspect SIP messages.

A comparison with SMTP [8] is instructive here. A much abused feature of SMTP is the "open relay" problem that is used to send email with a forged "From" header field. The same problem can occur in SIP as well. Consider that proxy.example.com establishes a TLS link with proxy.example.net. A cracker in the example.com domain sends a forged SIPS invitation to "sips:bob@example.net" claiming to be from "sip:alice@example.org". This message will make it successfully to the example.net domain, but clearly it has been forged and the UAS processing this should be able to notice the
discrepancy.

In SMTP, a common diagnostic tool is the "Received" field that contains an audit trail of the Mail Transfer Agents (MTA) that have handled the request thus far. The equivalent in SIP terms is the "Via" field. If the request actually originated in the example.org domain (remember, the message claimed to be from "sip:alice@example.org") then there should be a Via header corresponding to that domain. The absence of such a Via header is a hint that the message may have been compromised.

Thus, it seems appropriate to add a processing requirement to SIPS, which states that when a proxy at example.net receives a request claiming to be from proxy.example.com, then it should verify that proxy.example.com is the topmost Via entry in the Via header list. In addition, proxy.example.com should be the identity contained in the TLS certificate for the connection. Conversely, when proxy.example.net sends a SIPS message to example.com, the FQDN it adds in the sent-by production rule of the topmost Via header must be the same as the identity contained in the certificate it sent to example.com.

6. Security Considerations

This document is entirely concerned with security (more to be added).

7. Acknowledgments

The open issue in Section 5.4 was first documented by Klaus Darilion. Thanks to Cullen Jennings for graciously volunteering his time answering questions on the use of TLS in reSIProcate.

8. References

8.1 Normative References


8.2 Informative References


Authors’ Addresses

Vijay K. Gurbani
Lucent Technologies, Inc./Bell Laboratories
2701 Lucent Lane
Room 9F-546
Lisle, IL 60532
USA

Phone: +1 630 224-0216
Email: vkg at acm dot org

Alan S. Jeffrey
Lucent Technologies, Inc./Bell Laboratories
2701 Lucent Lane
Room 9F-534
Lisle, IL 60532
USA

Email: ajeffrey@lucent.com

Appendix A. TLS in SIP Test Cases
A.1 Tests for valid certificates

A.1.1 Good certificate, Case 1

Input: The server presents a certificate that is signed by a trusted certificate authority. The certificate has the canonical name of the host in the Distinguished Name (DN) of the Subject field. The subjectAltName X.509v3 extension is empty.

Expected behavior: Continue with the TLS session.

A.1.2 Good certificate, Case 2

Input: The server presents a certificate that is signed by a trusted certificate authority. The certificate has the canonical name of the host in the subjectAltName X.509v3 extension. The DN of the Subject field contains other identifying information besides a canonical name.

Expected behavior: Continue with the TLS session.

A.2 Tests for invalid certificates

A.2.1 Invalid certificate, Case 1

Input: The server presents a certificate that has expired.

Expected behavior: The client must immediately close the connection to the server.

Some frameworks, such as OpenSSL, automatically run default checks on certificates. One test among these default tests is the test for expiration. The OpenSSL framework informs the programmer that a certificate failed the default checks. The programmer must then close the connection opened with that server.

A.2.2 Invalid certificate, Case 2

Input: The server presents a certificate that is signed by a trusted certificate authority. However, there isn’t any canonical name in either the DN of the Subject field, or in the subjectAltName X.509v3 extension.

Expected behavior: Depends on the implementation. A paranoid implementation may want to terminate the TLS session immediately. A more lenient implementation may continue with the session with the expectation that, at the very least, the traffic is encrypted.
A.2.3 Invalid certificate, Case 3

Input: The server presents a certificate that is signed by a trusted certificate authority. However, the certificate has the wrong canonical name of the host in the Distinguished Name (DN) of the Subject field. The subjectAltName X.509v3 extension is empty.

Expected behavior: It is recommended that the TLS session be terminated.

A.2.4 Invalid certificate, Case 4

Input: The server presents a certificate that is signed by a trusted certificate authority. However, the certificate has the wrong canonical name of the host in the subjectAltName X.509v3 extension. The Distinguished Name (DN) of the Subject field contains other identifying information besides a canonical name.

Expected behavior: It is recommended that the TLS session be terminated.

A.2.5 Invalid certificate, Case 5

Input: The server presents a certificate that is signed by an unknown certificate authority.

Expected behavior: It is recommended that the TLS session be terminated.

A.3 Certificate depth test

Open question: How deep do we want to allow certificate chains to go?
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.