The SIPSEC Uniform Resource Identifier (URI)
draft-gurbani-sip-sipsec-01

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 December 15, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Currently, in the Session Initiation Protocol (SIP), there does not exist any means for a user agent client (UAC) to signal to the destination user agent server (UAS) that an end-to-end secure channel is to be established. Instead, what is prevalent today in the protocol is a hop-by-hop security model, wherein intermediaries forward a request towards the destination without the UAC knowing
whether or not the intermediary behaved in a trusted manner (i.e., it did not, unknown to the UAC, downgrade the security of the downstream channel from the intermediary onwards). This document discusses the security properties of a hop-by-hop model; and in doing so, formulates requirements an for an end-to-end security model and a solution that satisfies these requirements.

Table of Contents

1. Terminology .................................................................................................................. 3
2. Introduction ...................................................................................................................... 3
3. The Current SIP Security Model ..................................................................................... 4
4. Requirements for End-to-End Security ........................................................................... 5
5. Lessons Learned from Other Protocols ......................................................................... 6
6. Securing SIP signaling End to End .................................................................................. 6
   6.1. Overview .................................................................................................................... 7
   6.2. URI Format ................................................................................................................ 9
   6.3. The CONNECT method ........................................................................................... 10
   6.4. UAC behavior .......................................................................................................... 11
   6.5. UAS behavior .......................................................................................................... 11
   6.6. Proxy behavior ......................................................................................................... 12
   6.7. Example call flow ..................................................................................................... 12
   6.8. Adherence to requirements ..................................................................................... 14
7. Security Considerations .................................................................................................. 15
8. IANA considerations ....................................................................................................... 15
   8.1. URI Scheme .......................................................................................................... 15
   8.2. CONNECT request .................................................................................................. 16
   8.3. Option tag ............................................................................................................... 17
9. Acknowledgments ........................................................................................................... 17
10. References ...................................................................................................................... 17
    10.1. Normative References .......................................................................................... 17
    10.2. Informational References ..................................................................................... 18
Authors’ Addresses ........................................................................................................... 19
Intellectual Property and Copyright Statements .................................................................. 20
1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

2. Introduction

Intermediaries (proxy servers) play a big role in the routing of SIP requests [2]. Often times this is due to the network realities of today: in many deployments, firewalls and network address translators (NAT) prevent two endpoints from communicating directly with one other. At other times, intermediaries are mandatory due to the underlying characteristic of the network that requires a request to traverse through a set of intermediaries as it is routed towards its destination.

Traditionally, proxies have been trusted in SIP. User agents trust their proxies to provide the rendezvous service with the recipient, but this trust does not extend to include proxies being privy to sensitive information contained in the signaling messages. Since there may be a number of proxies between a UAC and a UAS, if the signaling information was to travel in the clear across this proxy mesh, then sensitive information in signaling (user identities, master keys for media encryption, etc.) is available to all the intermediaries that aid in the routing of that request. In fact, if the signaling session consisted entirely of, say, text messages for the Instant Message service (IM) in SIP [14], the intermediary proxies will have full access to the contents of the IM session, possibly violating the privacy of the users.

While the endpoints may trust the proxies responsible for their domain, this trust may not extend to other proxies, and in fact, the endpoints trust the proxies in their domain and in other domains just enough to route the request towards the recipient. The question of trust is most relevant in P2PSIP [18], where there is very limited recourse, if any, against a malicious node that betrays this trust.

From the viewpoint of the user agent client, two security properties are essential when establishing a session: one, it should be able to authenticate the recipient directly, and two, it should be assured of the privacy of communications across all the intermediaries that participate in getting the request to the destination. However, these are violated when proxies are used. Since direct communication with the recipient may not be possible due to NATs, proxies must be
used. Hence, a client cannot authenticate the recipient directly, and furthermore, it has to trust that the proxy does not violate privacy by snooping into and saving information from the signaling headers traversing through it.

This document explores how to provide end-to-end security outlined in the above paragraph in SIP. We start by first by deriving a list of requirements and then tracking these towards a possible solution.

3. The Current SIP Security Model

To be sure, SIP has the notion of providing confidentiality, integrity, and peer authentication to the signaling data flowing through the intermediaries. Four techniques exist to provide these properties, though not all of the techniques provide all of the properties simultaneously.

The first mechanism is the use of the SIPS scheme. This mechanism provides confidentiality, message integrity, although authentication is not end-to-end, but rather hop-by-hop. SIPS uses Transport Layer Security (TLS [5] [6]) on each hop from the UAC to the proxy serving the domain of the UAS (from then onwards, the protocol leaves the protection of the "last hop" up to the proxy responsible for that hop. The understanding is that the proxy should provide at least the same level of security to this traffic as was accorded to it over the link that deposited the traffic to the proxy.)

There are a number of areas in RFC3261 where the use of the SIPS scheme is left ambiguous. Audet [17] catalogues these areas and provides clarifications and normative changes to [2] wherever appropriate. The ambiguities in RFC3261 on SIPS notwithstanding, recent SIP interoperability events have witnessed a surge of TLS implementations (at the 18th SIP interoperability event held in April 2006, 30 out of 73 implementations supported TLS).

A rather insidious characteristic of SIPS is that the entire system depends on the proper functioning of intermediaries. It is not end-to-end, but rather, hop-by-hop and implies transitive trust. Furthermore, even if a proxy chain accords TLS protection on its every hop, all proxies in that chain have access to the plaintext signaling information as it traverses through that proxy. For some uses of SIP, this may preclude sending keying information for Secure Real-Time Protocol (SRTP [3]) in signaling, even if the signaling is protected by TLS across every hop.

A second mechanism to provide confidentiality and peer verification in SIP is the use of Secure/Multipurpose Internet Mail Extensions
(S/MIME [11]). S/MIME provides confidentiality, end-to-end peer verification, and integrity of the S/MIME body only. However, operationally S/MIME has proved to be quite hard to deploy (not all users possess certificates rooted in a trusted certificate authority), and programmatically, it has proved equally hard to implement (at the same SIP interoperability event, there weren’t any S/MIME implementations). Another drawback of S/MIME has also been that it sends some of the headers in cleartext. Unfortunately, this is unavoidable as the intermediaries need these headers to perform routing functions.

A third mechanism, Authenticated Identity Body (AIB [13]), provides integrity and authentication properties to the endpoints (it can optionally provide confidentiality as well). AIB suffers from much the same drawbacks as S/MIME, however.

And finally, SIP Identity [15] exists to provide authentication of the sender of the request and integrity of portions of the message body. Because SIP Identity relies on domain certificates instead of end-user certificates, it does not suffer from the same drawbacks as S/MIME does.

4. Requirements for End-to-End Security

We now list the requirements that an end-to-end SIP security mechanism should provide. The first requirement is rather tongue in cheek, but was widely discussed on the SIP mailing list as the infamous "hopping bunny with a lock in its mouth" requirement. This requirement allegorically depicts the state of security in SIP today with SIPS scheme, namely that SIPS provides hop-by-hop security and even then, there is no privacy or guards against eavesdropping since the intermediaries have access to the cleartext signaling. It appears that the Hypertext Transfer Protocol (HTTP [7]) has trained the users of the technology to associate a locked padlock with a secure connection. While this may be appropriate for HTTP, it is not appropriate for SIP since the SIPS scheme depends on the intermediaries to not behave maliciously (surreptitiously save the data flowing through them), and the characteristics of SIP are such that both the signaling layer, and the media layer established by that signaling must be confidential.

- REQ 1 - Stop the bunny from hopping :-)
- REQ 2 - Signaling must be confidential from intermediaries, even if the intermediaries are used to route the request towards the destination.
REQ 3 - A new URI scheme may be proposed that has the properties of confidentiality of signaling and media in an end to end manner. User agents using this new scheme, and intermediaries handling this new scheme will then have the adequate expectations of the session being set up. These expectations are that the caller is connected to the intended party and the confidentiality of the signaling and media is assured as well.

5. Lessons Learned from Other Protocols

The IETF has successfully added security to other protocols such as HTTP and Simple Mail Transport Protocol (SMTP [8]) after the protocol was specified. It helps to look at these cases to determine whether we can use the same or similar ideas in applying end to end security in SIP.

RFC 2818 [9] secures HTTP through the use of a new URI scheme - HTTPS. The analogous scheme in SIP is SIPS.

RFC 2817 [10], on the other hand, secures HTTP in a manner that can be used a model to secure SIP data end-to-end, with intermediaries aiding in routing the request. In HTTP, there may be one or more proxy between the browser and the origin server. When end-to-end security is desired, the browser establishes a TCP tunnel between itself and the origin server, through the proxies. Once the TCP tunnel is established, the browser and the origin server perform a TLS handshake to exchange X.509 certificates, agree on a set of algorithms, and establish a set of cryptographic keys. Subsequent data between them is encrypted and carried over the tunnel established by the proxies. Even though the proxies transport the data, they cannot decrypt it or perform any analysis on it.

RFC 3207 [12] specifies the STARTTLS service, which negotiates a TLS connection once a TCP connection has been set up between a SMTP client and an SMTP server.

6. Securing SIP signaling End to End

This document proposes a mechanism similar to the HTTP-tunneling mechanism to establish an end-to-end secure overlay transport path across a set of SIP proxies. We first explain the mechanism itself, including the SIP extensions that are needed to realize it, and then look at how the mechanism fulfills the requirements listed in Section 4.
6.1. Overview

This subsection contains an example flow of an end-to-end secure session using the "sipsec" URI scheme (see Section 6.2). Figure 1 shows an overall time-line diagram of using the "sipsec" URI scheme to establish a communication with the destination UAS through a mesh of intermediaries.

```
UAC       Proxy 1       Proxy 2       UAS
              +-----------+             |
              |TCP hand-  |             |
              |shake      |             |
              +-----------+             |
              |           |             |
              |CONNECT    |             |
              +-----------+             |
              |TCP hand-  |             |
              |shake      |             |
              +-----------+             |
              |           |             |
              |CONNECT    |             |
              +----------->             |
          200 OK     |             |
          -----<------|<------------------
```

Figure 1: Initial rendezvous

The UAC, on being presented with a sipsec URI, creates a request with a method of CONNECT. This request has a R-URI of the destination UAS. The request is routed downstream as specified in [2]. The transport used for the request on each hop must be a connection-oriented, congestion-controlled transport; in this example, it is TCP. When the UAS receives the request, it denotes its desire to establish a session with the sender using a 200-class response to the CONNECT request. This response is sent back over the same transport...
that the request arrives on.

This specification does not prohibit nor does it explicitly mandate a set of headers to appear in the CONNECT request. Thus, the sender can add any legal SIP header (such as a From) to the request. The desire for privacy may drive the inclusion (or exclusion) of certain headers when the UAC creates a request. Similarly, intermediaries can challenge the CONNECT request or add any header pertinent to their operation if it is permitted by SIP. Section 6.3 contains normative text on the use of the CONNECT request.

Once this rendezvous has been made, the intermediaries put themselves into a "transparent" mode. While in this mode, any data they receive from their upstream neighbor will be sent to the downstream neighbor, and any data received from the downstream neighbor will be sent to the upstream neighbor.

Figure 2 depicts the situation after the rendezvous is complete. Here, the UAC and the UAS upgrade the connection to use TLS; X.509 certificates are exchanged and encryption keys are derived. The certificates enable both the end points to identify themselves to each other and the encryption keys allow the subsequent application data to be rendered opaque to the intermediaries.
When the connection is no longer needed, either of the end points can issue a TLS shutdown. This shutdown will be followed by a per-hop TCP disconnect.

6.2. URI Format

End to end secure SIP is differentiated from hop-by-hop secure SIP (SIPS URI) by using the "sipsec" protocol scheme identifier in place of the "sips" protocol scheme identifier. An example URI specifying the need for an end-to-end secure SIP is:
sipsec:vkg@example.com

However, it is not expected that the sipsec URI scheme will be serialized on the wire; instead, the semantics accorded to it are in the same vein as a mailto URI wherein a browser, on seeing the mailto URI presents a mail client to the user. Similarly, a SIP user agent on being presented a sipsec URI MUST formulate a SIP CONNECT request with a SIP or SIPS URI scheme. The SIP user agent MUST also insert the "sipsec" option tag in the Proxy-Require and Require header. The normative behavior of intermediaries, user agent clients and user agent servers on encountering this option tag is presented in subsequent sections of this document.

6.3. The CONNECT method

This document requests the IANA to register a new SIP request method called CONNECT. The CONNECT method and its corresponding 200-class response is used by user agents to indicate the desire of an end-to-end secure session. Tables 1 and 2 below extend Tables 2 and 3, respectively, from [2].

TO DO: Provide Tables 1 and 2.

At the minimum, this specification requires a request with a CONNECT method MUST contain the Request-Line, Proxy-Require header, Require header and CRLF as defined in the ABNF of [2]. The Proxy-Require and Require headers MUST contain the option tag "sipsec". Any other SIP headers or body types are permissible if they are allowed by the protocol and if the UAC sending the request wishes to use them.

Under some circumstances, a UAC may not wish to divulge any other information about itself beyond the fact that a CONNECT request was made to a certain UAS. If the request is successfully answered, the rest of the signaling messages will be encrypted end to end. Thus, an intermediary will know that a CONNECT request was established, but what transpired after that remains private.

The CONNECT request, like any other SIP request, may be challenged by a proxy upon which the user will have to provide appropriate credentials. Similarly, if the user already possesses a X.509 certificate, it may use TLS to establish a connection with the proxy.

At the minimum, this specification requires that a response to a CONNECT request MUST contain the Status-Line and CRLF as defined in the ABNF of [2]. Any other SIP headers or body types are permissible if they are allowed by the protocol and if the UAS sending the response wishes to use them. Privacy concerns outlined above to requests apply to responses as well.
6.4. UAC behavior

A UAC that wishes to use end-to-end encryption MUST first formulate a new request with the method name CONNECT. The Request-URI (R-URI) scheme MAY be left as a SIP URI or it may be a SIPS URI. Regardless, no other information is required for the CONNECT request. The CONNECT request SHOULD NOT contain a payload.

For some communications, the caller may not want to reveal any private information at all. The CONNECT request may travel in the clear across a list of proxies until it arrives at the intended destination identified in the R-URI. Thus, limiting the amount of information available to intermediaries is a good strategy. Of course, if a UAC has no compunctions on revealing private information, then it may add any kind of information that it wants to make available to the receiving endpoint, and by extension, to each of the intermediaries fielding this request.

The UAC MUST include a Proxy-Require and a Require header in the CONNECT request with an option tag value of "sipsec".

When a corresponding 200-class response arrives for the CONNECT request, the UAC can start the TLS handshake, and once the handshake is done, actual data can follow. The handshake and data will traverse over the overlay TCP network established by the intermediaries, however, since the encryption keys were exchanged end-to-end, the intermediaries will not be able to eavesdrop on the communications.

The data in this case consists of another SIP request that sets up the actual communication channel between the UAC and the UAS. However, before sending this new SIP request, the UAC MUST examine the X.509 certificate presented to it during the handshake to determine the veracity of the UAS. If the UAC does not trust the identity of the UAS based on the information contained in the certificate, it MUST present this information to the user associated with the UAC and allow it to make a decision on whether to continue with the connection or to abort the attempt.

6.5. UAS behavior

A UAS that wishes to use end-to-end encryption should inspect an incoming CONNECT request for a Require header field containing an option value of "sipsec". If such a header is found, and the UAS is willing to enter a communication session with the sender (at this point, the UAS may not know who the sender is), it MUST send a 200-class response to the CONNECT and prepare to accept a TLS handshake from the sender.
To ascertain the identity of the sender of the request, the UAS MUST configure its TLS library such that it asks for an X.509 certificate from the sender during the TLS handshake. All commonly used TLS libraries allow the server to ask the client of its certificate during the initial handshake (for example, in the OpenSSL library, the SSL_VERIFY_PEER flag can be set for the server SSL context; when this flag is set, the server will ask the client for its certificate.) If the UAS does not trust the sender of the CONNECT request based on the identity contained in the certificate, it MUST present this information to the user associated with the UAS and allow it to make a decision on whether to continue with the connection or to abort the attempt. If the connection setup continues, the UAS MUST be prepared to receive a SIP request that will indicate the type of communications the sender wishes to set up.

6.6. Proxy behavior

A proxy, on the receipt of a new CONNECT request must inspect the Proxy-Require header field and if it contains the option tag "sipsec" then the proxy MUST route the request downstream as specified in [2]. Upon receiving a corresponding 200-class response to the request, the proxy MUST send the response upstream as specified in [2]. Then, it MUST leave the TCP connection over which the request and response traversed open, and prepare to enter into a mode where it forwards the data arriving on this connection from the sender to the receiver. Because the data arriving on this connection after the 200 OK has been sent upstream may be encrypted, the proxy will not be able to interpret it. However, it MUST keep the TCP connection open until the connection is explicitly torn down by either of the endpoints. In extreme conditions, if the proxy has not observed any data on this connection for some time period (configurable per proxy with a minimum default value of 1800s MUST be used), it MAY actively shut down the TCP connection.

TODO: Clarify interaction with SIP Session Timer [16].

6.7. Example call flow

A representative call flow diagram is provided in Figure 3 below.
Figure 3: Example call flow

In this call flow, the UAC negotiates a path to the UAS through two
proxies, P1 and P2 (steps 1 - 3). After the UAS has been contacted and has acquiesced to communication (step 4), the intervening proxies put themselves in cryptographically transparent mode (steps 6 and 8) and essentially establish a virtual transport path (a tunnel) from the UAC to the UAS going through the proxies (step 9). The proxies, while in this mode, will receive data -- which to them is simply an opaque sequence of bytes -- on one side of the transport connection and send it to the other side.

On the establishment of this tunnel, the UAC initiates a TLS handshake directly to the UAS (step 10). The UAS responds with its certificate and the keying material for confidentiality is derived, and the endpoints are authenticated as per the properties of the certificates exchanged (i.e., a self-signed certificate may be validated through a fingerprint exchanged between the UAS and the UAC some time before, or a certificate signed by a trusted third party is verified by following the trust chain.) For mutual authentication, both sides of the virtual connection must provide X.509 certificates (steps 11-15).

After the tunnel has been authenticated and keying material derived to secure the confidentiality, application layer messaging follows between the UAC and UAS (steps 16, 18, 19.) In Figure 3, the application layer messaging is composed of the INVITE transaction. Note that once the request has been sent to the UAS, the UAS may have to undertake more trust decision based on the nature of signaling (for example, a user associated with the UAC may want to subscribe to some resource of a user associated with a UAS, but the user of the UAS does not want information for that resource disseminated to the UAC despite the fact that the user associated with the UAC has been authenticated. In other words, once the identities of the endpoints have been asserted and the confidentiality of the channel protected, application-specific trust decision may still be made.)

6.8. Adherence to requirements

In this section, we evaluate the sipsec URI scheme to the requirements we captured in Section 4.

- REQ 1 - Stop the bunny from hopping :-) ‘Nuff said.
- REQ 2 - Signaling must be confidential from intermediaries, even if the intermediaries are used to route the request towards the destination. This requirement is met as depicted in Figure 2.
- REQ 3 - A new URI scheme may be proposed that has the properties of confidentiality of signaling and media in an end to end manner. User agents using this new scheme, and intermediaries handling this new scheme will then have the adequate expectations of the session being set up. These expectations are that the caller is
connected to the intended party and the confidentiality of the
signaling and media is assured as well. This requirement has been
met with the introduction of the "sipsec" URI scheme (see
Section 6.2).

7. Security Considerations

Even though this entire draft pertains to security in general, the
manner by which certain components are employed to provide the end-
to-end security bears some more scrutiny.

The initial request to establish an end-to-end tunnel -- the CONNECT
request -- requires the R-URI of the recipient to be provided for
routing purposes. Thus, at the very least, the identity (or one of
the identities) of the recipient is conceivably divulged.

The scheme described in this document does depend on X.509
certificates. In not all of the instances will user agents have such
a certificate rooted in a trusted hierarchy and issued by a trusted
certificate authority. In some cases, user agents may use self-
signed certificates. However, even the use of self-signed
certificates provides confidentiality for endpoint communication
across a set of proxies that would not otherwise exist. A self-
signed certificate cannot possibly be used for authoritative
authentication unless both parties have exchanged certificate
fingerprints a-priori over another channel (face-to-face meeting,
postal service, email, etc.)

8. IANA considerations

IANA considerations for registering a new URI scheme, a new SIP
method, and a new SIP option tag are provided in the subsequent
subsections, respectively.

8.1. URI Scheme

Following the directions of [4], the following template is provided
to register the URI scheme:

URI scheme name: sipsec

Status: permanent
URI scheme syntax:

```
SIPSEC-URI = "sipsec:" [ userinfo ] hostport
               uri-parameters [ headers ]
```

userinfo, hostport, uri-parameters, and headers production rules are defined in [2].

URI scheme semantics: See Section 6.2.

Encoding considerations: There are no special encoding considerations.

Applications/protocols that use this URI scheme name: SIP

Interoperability considerations: A SIP user agent client, when presented with this scheme, will process it as described in this document; if it does not support the scheme, will display a "416 Unsupported URI scheme" to its user. Intermediary and endpoint support is indicated by the presence of a new option tag being registered by this document.

Security considerations: See Section 7

Contact: Vijay K. Gurbani, vkg at acm.org

Author/Change controller: The IESG.

References: None beyond what are specified in this document.

8.2. CONNECT request

This document requests IANA to register the new SIP method CONNECT at the SIP Registry (http://www.iana.org/assignments/sip-parameters) as follows:

```
Methods       Reference
------------   --------
CONNECT       [RFCXXXX]
```

RFC Editor: Please replace RFCXXXX with the number assigned to this document.
8.3. Option tag

This document requests IANA to register the option tag "sipsec" at the SIP Registry (http://www.iana.org/assignments/sip-parameters) as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>sipsec</td>
<td>This option tag is for indicating support for an end-to-end X.509 certificate exchange in SIP to provide confidentiality and authentication.</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

RFC Editor: Please replace RFCXXXX with the number assigned to this document.

An example of the use of this tag is below:

```
Proxy-Require: sipsec
or
Require: sipsec
```

This tag is used in the Require and Proxy-Require headers.

9. Acknowledgments

The following people contributed immensely to the discussion on the SIP working group mailing list: Jeroen van Bemmel (suggested using "SIPSEC" as the name for the URI scheme as opposed to the originally envisioned name of "SIPSS"), Michael Hammer (suggested the "hopping bunny" allegory), Juha Heinanen, Christer Holmberg, Alan Jeffrey, Cullen Jennings, Paul Kyzivat, Rohan Mahy, Frank Miller, Aki Niemi, Brian Rosen, Jonathan Rosenberg, Michael Thomas, and Dale Worley.

10. References

10.1. Normative References


10.2.  Informational References


Authors’ Addresses

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
2701 Lucent Lane
Rm 9F-546
Lisle, IL  60532
USA
Phone:  +1 630 224 0216
Email:  vkg@alcatel-lucent.com

Francois Audet
Nortel Networks
4655 Great America Parkway
Santa Clara, CA  95054
USA
Phone:  +1 408 495 2456
Email:  audet@nortel.com

Dean Willis

Email:  dean.willis@softarmor.com