Network Working Group
Request for Comments: 4189
Category: Informational

K. Ono
S. Tachimoto
NTT Corporation

October 2005

Requirements for End-to-Middle Security for the Session Initiation Protocol (SIP)

Status of This Memo

Copyright Notice

Abstract

A Session Initiation Protocol (SIP) User Agent (UA) does not always trust all intermediaries in its request path to inspect its message bodies and/or headers contained in its message. The UA might want to protect the message bodies and/or headers from intermediaries, except those that provide services based on its content. This situation requires a mechanism called "end-to-middle security" to secure the information passed between the UA and intermediaries, which does not interfere with end-to-end security. This document defines a set of requirements for a mechanism to achieve end-to-middle security.

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP) [2] supports hop-by-hop security using Transport Layer Security (TLS) [3] and end-to-end security using Secure MIME (S/MIME) [4]. Use of TLS assumes that a SIP UA trusts all proxy servers along its request path to inspect the message bodies contained in the message, and use of S/MIME assumes that a SIP UA does not trust any proxy servers to do so.

However, there is a model in which trusted and partially-trusted proxy servers are mixed along a message path. The partially-trusted proxy servers are only trusted to provide SIP routing, but these proxy servers are not trusted by users to inspect its data, except the routing headers. A hop-by-hop confidentiality service using TLS is not suitable for this model. An end-to-end confidentiality service using S/MIME is also not suitable when the intermediaries provide services based on reading the message bodies and/or headers. This problem is described in Section 23 of [2].

In some cases, a UA might want to protect its message bodies and/or headers from proxy servers along its request path, except from those that provide services based on reading its message bodies and/or headers. Conversely, a proxy server might want to view the message bodies and/or headers to sufficiently provide these services. Such proxy servers are not always the first hop from the UA. This situation requires a security mechanism to secure message bodies and/or headers between the UA and the proxy servers, while disclosing information to those that need it. We call this "end-to-middle security".

1.1. Conventions Used in This Document

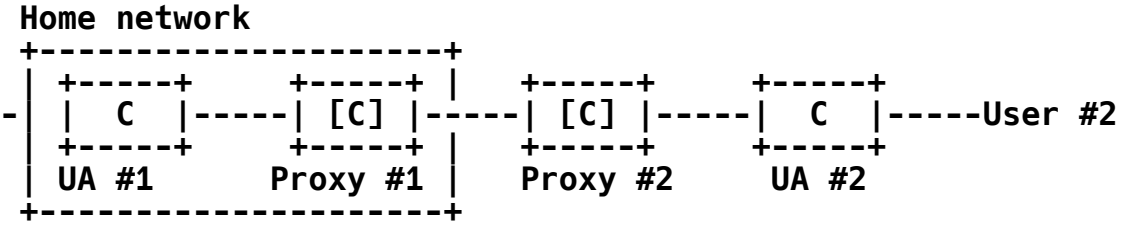
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

2. Use Cases

2.1. Examples of Scenarios

We describe here examples of scenarios in which trusted and partially-trusted proxy servers both exist in a message path. These situations demonstrate the reasons why end-to-middle security is required.

In the following example, User #1 does not know the security policies or services provided by Proxy server #1 (Proxy#1). User #1 sends a MESSAGE [5] request including S/MIME-encrypted message content for end-to-end security, as shown in Figure 1, while Proxy #1 rejects the request based on its strict security policy that prohibits the forwarding of unknown data.

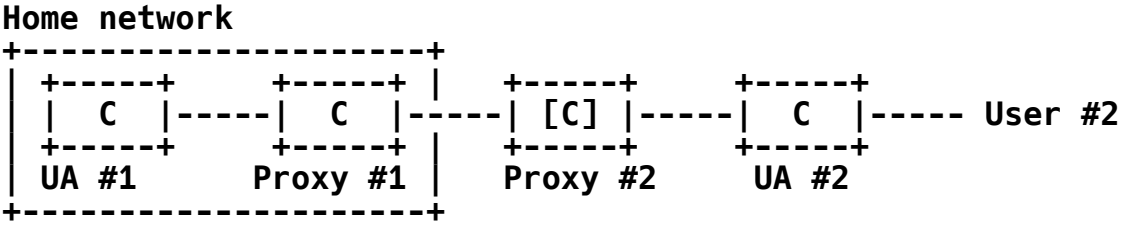


C: Content that UA #1 allows the entity to inspect
[C]: Content that UA #1 prevents the entity from inspecting

Figure 1: Deployment example #1

In the second example, Proxy server #1 is the home proxy server of User #1 using UA #1. User #1 communicates with User #2 through Proxy #1 and Proxy #2, as shown in Figure 2. Although User #1 already knows Proxy #1's security policy, which requires the inspection of the content of the MESSAGE request, User #1 does not know whether Proxy #2 is trustworthy, and thus wants to protect the message bodies in the request. To accomplish this, UA #1 will need to be able to grant a trusted intermediary (Proxy #1) to inspect message bodies, while preserving their confidentiality from other intermediaries (Proxy #2).

Even if UA #1's request message authorizes Proxy #1 to inspect the message bodies, UA #1 is unable to authorize the same proxy server to inspect the message bodies in subsequent MESSAGE requests from UA #2.

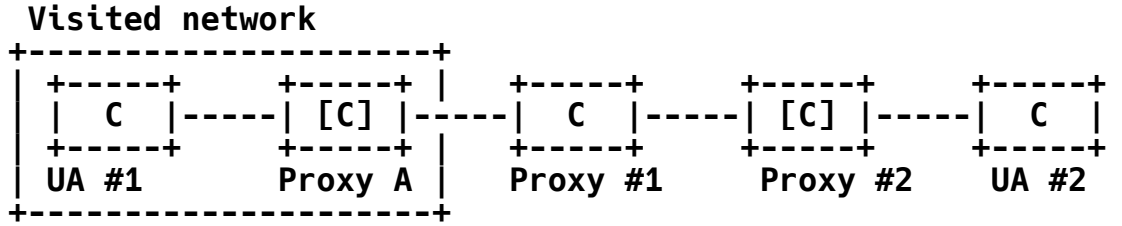


C: Content that UA #1 needs to disclose
[C]: Content that UA #1 needs to protect

Figure 2: Deployment example #2

In the third example, User #1 connects UA #1 to a proxy server in a visited (potentially insecure) network, e.g., a hotspot service or a roaming service. Since User #1 wants to utilize certain home network services, UA #1 connects to a home proxy server, Proxy #1. However, UA #1 must connect to Proxy #1 via the proxy server of the visited network (Proxy A), because User #1 must follow the policy of that network. Proxy A performs access control based on the destination addresses of calls. User #1 only trusts Proxy A to route requests, not to inspect the message bodies the requests contain, as shown in Figure 3. User #1 trusts Proxy #1 both to route the requests and to inspect the message bodies.

The same problems as in the second example also exist here.



C: Content that UA #1 needs to disclose
[C]: Content that UA #1 needs to protect

Figure 3: Deployment example #3

2.2. Service Examples

We describe here several services that require end-to-middle security.

2.2.1. Logging Services for Instant Messages

Logging Services are provided by the archiving function, which is located in the proxy server, that logs the message content exchanged between UAs. The archiving function could be located at the originator network and/or the destination network. When the content of an instant message contains private information, UACs (UA Clients) encrypt the content for the UASes (UA Servers). The archiving function needs to log the content in a message body in bidirectional MESSAGE requests in such a way that the data is decipherable. The archiving function also needs a way to verify the data integrity of the content before logging.

This service might be deployed in financial networks, health care service provider's networks, as well as other networks in which archiving communication is required by their security policies.

2.2.2. Non-emergency Call Routing Based on the Location Object

The Location Object [6] includes a person's geographical location information that is privacy-sensitive. Some proxy servers will have the ability to provide routing based on the geographical location information. When UAs want to employ location-based routing in non-emergency situations, the UAs need to connect to the proxy servers with such a capability and disclose the geographical location information contained in the message body of the INVITE request, while protecting it from other proxy servers along the request path. The Location Object also needs to be verified for data integrity by the proxy servers before location-based routing is applied. Sometimes the UACs want to send the Location Object to the UASes. This is another good example that presents the need for UACs to simultaneously send secure data to a proxy server and to the UASes.

2.2.3. User Authentication

2.2.3.1. User Authentication Using the AIBs

The Authenticated Identity Bodies (AIBs) [7] is a digitally-signed data that is used for identifying users. Proxy servers that need to authenticate a user, verify the signature. When the originator needs anonymity, the user identity in the AIB is encrypted before being signed. Proxy servers that authenticate the user need to decrypt the body in order to view the user identity in the AIB. Such proxy servers can be located adjacently and/or non-adjacently to the UA.

The AIB could be included in all request/response messages. The proxy server needs to view it in request messages in order to authenticate users. Another proxy server sometimes needs to view it in response messages for user authentication.

2.2.3.2. User Authentication in HTTP Digest Authentication

User authentication data for HTTP Digest authentication [8] includes potentially private information, such as a user name. The user authentication data can be set only in a SIP header of request messages. This information needs to be transmitted securely to servers that authenticate users, located either adjacently and/or non-adjacently to the UA.

2.2.4. Media-related Services

Firewall traversal is an example of services based on media information in a message body, such as the Session Description Protocol (SDP) [9]. A firewall entity that supports the SIP protocol, or a midcom [10] agent co-located with a proxy server,

controls a firewall based on the address and port information of media streams in the SDP offer/answer. The address and port information in the SDP needs to be transmitted securely to recipient UAs and the proxy server operating as a midcom agent. Therefore, there is a need for a proxy server to be able to decrypt the SDP, as well as to verify the integrity of the SDP.

When the SDP includes key parameters for Secure RTP (SRTP) [11], the key parameters need to be encrypted only for end-to-end confidentiality.

3. Scope of End-to-Middle Security

End-to-middle security consists of user authentication, data integrity, and data confidentiality. Providing data integrity requires authenticating peer who creates the data. However, this document only describes requirements for data confidentiality and data integrity, since end-to-middle authentication is covered by existing mechanisms such as HTTP Digest authentication, S/MIME Cryptographic Message Syntax (CMS) SignedData body [12], or an AIB.

As for data integrity, the CMS SignedData body can be used for verification of the data integrity and authentication of the signer by any entities. The CMS SignedData body can be used for end-to-middle security and end-to-end security simultaneously. However, a proxy server generally does not verify the data integrity using the CMS SignedData body, and there is no way for a UA to request the proxy server to verify the message. Therefore, some new mechanisms are needed to achieve data integrity for end-to-middle security.

This document mainly discusses requirements for data confidentiality and the integrity of end-to-middle security.

4. Requirements for a Solution

We describe here requirements for a solution. The requirements are mainly applied during the phase of a dialog creation or sending a MESSAGE request.

4.1. General Requirements

The following are general requirements for end-to-middle confidentiality and integrity.

REQ-GEN-1: The solution SHOULD have little impact on the way a UA handles S/MIME-secured messages.

REQ-GEN-2: It SHOULD NOT have an impact on proxy servers that do not provide services based on S/MIME-secured bodies in terms of handling the existing SIP headers.

REQ-GEN-3: It SHOULD NOT violate the standardized mechanism of proxy servers in terms of handling message bodies.

REQ-GEN-4: It SHOULD allow a UA to discover security policies of proxy servers. Security policies imply what data is needed to disclose and/or verify in a message.

This requirement is necessary when the UA does not know statically which proxy servers or domains need disclosing data and/or verification.

4.2. Requirements for End-to-Middle Confidentiality

REQ-CONF-1: The solution MUST allow encrypted data to be shared with the recipient UA and a proxy server, when a UA wants.

REQ-CONF-2: It MUST NOT violate end-to-end encryption when the encrypted data does not need to be shared with any proxy servers.

REQ-CONF-3: It SHOULD allow a UA to request a proxy server to view specific message bodies. The request itself SHOULD be secure; namely it SHOULD be authenticated for the UA and verified for the data integrity.

REQ-CONF-4: It MAY allow a UA to request that the recipient UA disclose information to the proxy server to which the requesting UA is initially disclosing information. The request itself SHOULD be secure; namely it SHOULD be authenticated for the UA and verified for the data integrity.

This requirement is necessary when a provider operating the proxy server allows its security policies to be revealed to the provider serving the recipient UA.

4.3. Requirements for End-to-Middle Integrity

This section enumerates the requirements for the end-to-middle integrity. Verifying the data integrity requires checking that the data is created by the authenticated user and not forged by a malicious user. Therefore, verification of the data integrity requires the user authentication.

REQ-INT-1: The solution SHOULD work even when the SIP end-to-end authentication and integrity services are enabled.

REQ-INT-2: It SHOULD allow a UA to request a proxy server to verify specific message bodies and authenticate the user. The request itself SHOULD be secure; namely it SHOULD be authenticated for the UA and verified for the data integrity.

REQ-INT-3: It SHOULD allow a UA to request the recipient UA to send the verification data of the same information that the requesting UA is providing to the proxy server. The request itself SHOULD be secure; namely it SHOULD be authenticated for the UA and verified for the data integrity.

This requirement is necessary when a provider operating the proxy server allows its security policies to be revealed to the provider serving the recipient UA.

5. Security Considerations

This document describes the requirements for confidentiality and integrity between a UA and a proxy server. Although this document does not cover any requirements for authentication, verifying the data integrity requires peer authentication. Also, peer authentication is important in order to prevent attacks from malicious users and servers.

The end-to-middle security requires additional processing on message bodies, such as unpacking MIME structure, data decryption, and/or signature verification to proxy servers. Therefore, the proxy servers that enable end-to-middle security are vulnerable to a Denial-of-Services attack. A threat model is where a malicious user sends many complicated-MIME-structure messages to a proxy server, containing user authentication data obtained by eavesdropping. Another threat model is where a malicious proxy server sends many complicated-MIME-structure messages to a proxy server, containing the source IP address and the Via header of an adjacent proxy server. These attacks will slow down the overall performance of target proxy servers.

To prevent these attacks, user and server authentication mechanisms need to be protected against replay attacks, or the user and server authentication always need to be executed simultaneously with protection of data integrity. In order to prevent these attacks, the following requirements should be met.

- The solution MUST support mutual authentication, data confidentiality, and data integrity protection between a UA and a proxy server.
- It SHOULD support protection against a replay attack for user authentication.
- It SHOULD simultaneously support user authentication and data integrity protection.

  These last two requirements are met by HTTP Digest authentication.

- It MUST support mutual authentication, data confidentiality, and data integrity protection between proxy servers.
- It SHOULD support protection against a replay attack for server authentication.
- It SHOULD simultaneously support server authentication and data integrity protection.

  These last three requirements are met by TLS.

6. Acknowledgments

The authors would like to thank to Rohan Mahy and Cullen Jennings for their initial support of this concept, and to Jon Peterson, Gonzalo Camarillo, Sean Olson, Mark Baugher, Mary Barnes, and others for their reviews and constructive comments.

7. References

7.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] 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.

7.2. Informative References

[3] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[4] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling", RFC 3850, July 2004.

[5] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[6] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, October 2005.

[7] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.

[8] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[9] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[10] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[11] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[12] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

Authors' Addresses

Kumiko Ono
Network Service Systems Laboratories
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-shi, Tokyo 180-8585
Japan

EMail: ono.kumiko@lab.ntt.co.jp, kumiko@cs.columbia.edu

Shinya Tachimoto
Network Service Systems Laboratories
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-shi, Tokyo 180-8585
Japan

EMail: tachimoto.shinya@lab.ntt.co.jp

Full Copyright Statement

Intellectual Property

Acknowledgement

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