SIP
Internet-Draft
Expires: July 14, 2006

F. Cao, Ed.
Cisco Systems

January 10, 2006

Response Authentication in Session Initiation Protocol
draft-cao-sip-response-auth-01

Status of this Memo

Copyright Notice

Abstract

This draft describes some extensions for enhancing SIP response authentication. In the real-world SIP deployment, TLS may be not available on some hops. Due to the lack of other response authentication mechanisms in SIP, several kinds of security attacks could be conducted on those hops through SIP response. This draft suggests some approaches for complementary enhancement on SIP response authentication. With the new per-hop response authentication proposed in this draft, the security gaps on the hops without TLS can be bridged.

Table of Contents

1. Introduction

This document provides complementary enhancements for addressing security concerns on response authentication in Session Initiation Protocol (SIP [1]). [3] described the current limitations of some security mechanisms provided in SIP ([1]). One of them is about the difficulties of deploying TLS over each hop of all SIP dialogs. Because SIPs is the only mechanism for response authentication, the lack of TLS on some hops imposes some threats of malicious attacks through SIP response to disturb the desired service. In particular, there is no strict per-hop authentication for the received SIP response when TLS is absent. This may enable the attackers to spoof SIP response and easily disturb the SIP service.

For example, if a rogue proxy can sniff the SIP requests from Proxy-1 to Proxy-2 without TLS, it can spoof the addresses and URIs of Proxy-2 and fake the response back to Proxy-1 along with its own rogue domain authentication service info, right before Proxy-2's response. Proxy-1 and the initiators of SIP requests will be deceived by the responses from the rogue proxy. This allows the rogue proxy to conduct many attacks, such as redirecting the requests to attack other targets for DoS attacks, redirecting the requests to rogue users for information disclosure, and terminating the requests for turning down SIP services.

This draft suggests some approaches for complementary enhancement on per-hop response authentication inside SIP, which can bridge the gap when TLS is absent on those hops.

2. 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 [2].

Domain-based Authentication Service (DAS): Authentication service is provided for each domain through its certificate and the domain private key. Proxies may act as the role of authenticate service with the domain private keys.

Authenticated Identity Body (AIB): some SIP headers are replicated into a S/MIME body of the same message and are signed with a digital signature (See [5])

Chain of SIP Response Trust (CSRT): All the hops in the path of SIP response provides the authentication mechanisms so that the chain of the trust on the response message can be built from end to end.

Certificate: An X.509v3 [15] style certificate containing a public key and a list of identities in the SubjectAltName that are bound to this key. The certificates discussed in this document are generally self signed and use the mechanisms in the SIP Identity specification to vouch for their validity.

3. Overview

This section gives an overview of the requirements and the mechanisms for addressing the security concerns of SIP response. In particular, some security mechanisms on per-hop authentication are proposed to enhance the response authentication and prevent the malicious attacks through SIP response.

The following requirements should be addressed:

- The new response authentication must be complementary with SIPs, i.e. it should work when TLS is absent.
- Response authentication between neighboring domains or nodes can be enhanced
- The mechanism should be simple
- CSRT can be built when either this mechanism is applied on all the hops, or this mechanism is applied on some of the hops and TLS is used for the rest.

3.1. Per-hop Authentication Enhancement

One simple authentication mechanism is proposed in this document for satisfying all these requirements. This mechanism is to generate a digest challenge for the next-hop node (or domain), and the authorization to this challenge should be delayed, and piggybacked with the next normal SIP response from the next-hop downstream node (or domain). After the digest is verified, the trust can be enhanced for the SIP response from the next-hop node (or domain).

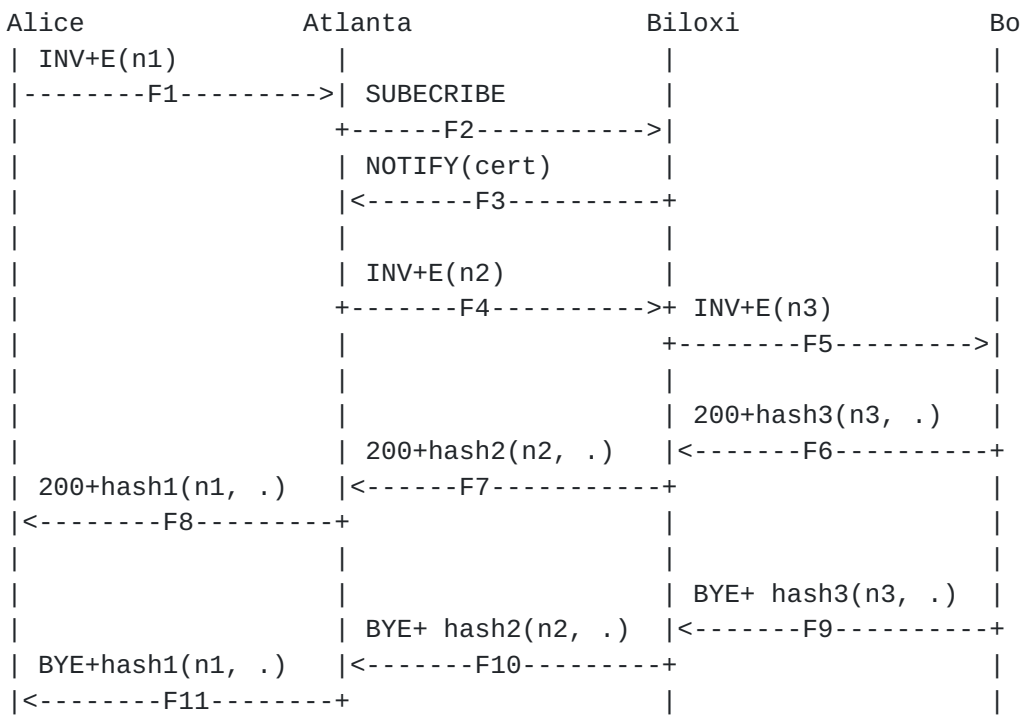
There are several security mechanisms covered in this document to support this mechanism:

O DAS

O shared secret key with the next-hop downstream node

O public key of the next-hop downstream node

The figure below shows a basic call to illustrate some scenarios. The call is initiated by alice@atlanta.com to bob@biloxy.com. The assumption is that Alice and Atlanta have a shared secret, Biloxi has a public certificate, and Bob and Biloxi have a shared secret.



In message F1, Alice sends a normal invite but includes an Authentication header that include the encrypted nonce, n1, that is encrypted for the next hop which is Atlanta.

In message F4, Atlanta will forward the invite to Bilboxi with a nonce that is encrypted for Biloxi however, to do the encryption, Atlanta may have to use the Sub/NOT if message F2 and F3 to fetch Biloxi's public key so that it can encrypt the nonce. Note F2 and F3 might be done for previous SIP dialogs from Atlanta.com to Bilboxi.com.

In message F5, biloxi sends the INVITE with a nonce encrypted for bob using the shared secret between Biloxi and Bob.

In message F6, Bob inserts a header that says the responder in bob@biloxi.com and computes a hash over key parts of the message including the responder header field value. The hash includes the decrypted content of the nonce that Biloxi sent to Bob. When biloxi receives this message it can verify that it the hash is correct and that it believes the responder information.

Biloxi computes a new hash over the message using the nonce2 and sends F7 using this hash.

Later in message F9, F10, and F11, the hash can be computed using the previous nonces. The proxies do not need to be session state-full as long as the nonce are constructed in a way such that the proxy can

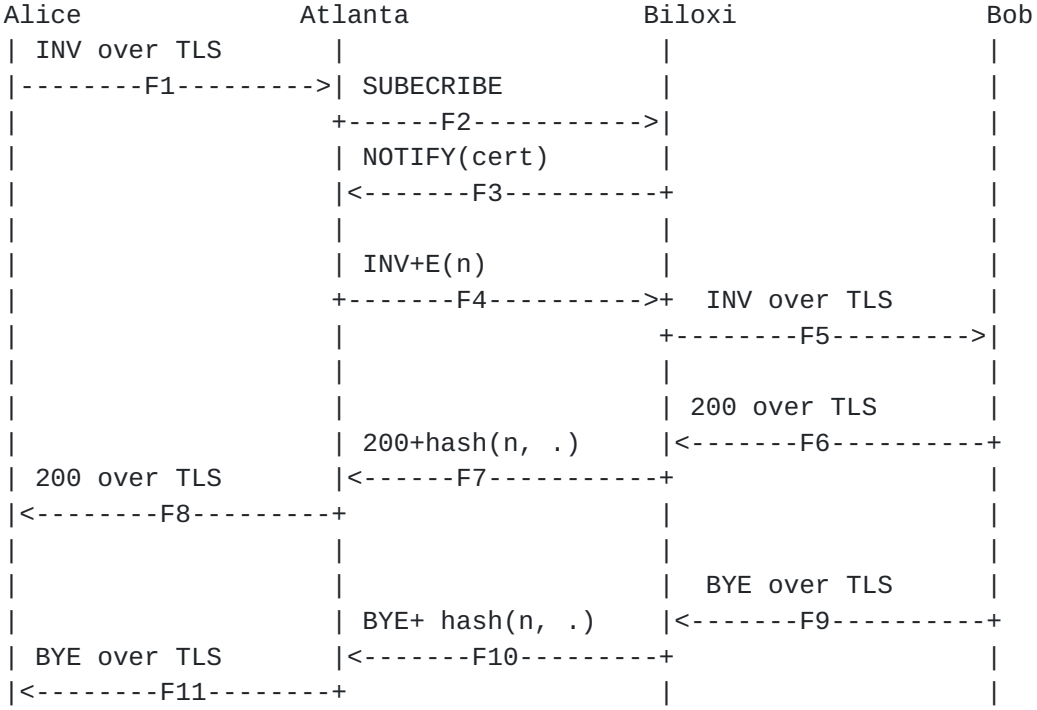
later check that they are only being used in the dialog for which they were originally constructed.

If the verification in Biloxy or Atlanta indicates the unmatched SIP response authorization, the proxy may replace the response code with 432 Failed Response Authorization for announcing the failure of the next-hop response authentication.

3.2. Complementariness with TLS

This proposed per-hop authentication mechanism is complementary with TLS in SIP deployment. If TLS is available on some hops, this mechanism can be applied to the other hops where TLS is absent. The below example demonstrate how they work together.

Assume that TLS is available between Alice and Atlanta AND between Biloxi and Bob. The hop between Atlanta and Biloxi doesn't support TLS.



TLS can be applied to create the security tunnels for two hops, i.e. between Alice and Atlanta AND between Biloxy and Bob. Similar to the above example, the hop between Atlanta and Biloxy can use the proposed response authentication mechanism to enhace response security.

All the hops in this example provide the security mechanisms to check

that
o the received response message from the desired upstream node
o the integrity of this received message can be verified.

Therefore, CSRT can be built by combining this extension and TLS from end to end. The rogue proxies can be prevented from attacking SIP services through SIP responses.

4. User Agent Behavior

The extensions in this document require new processing and parsing for both UAS and UAC. Their behaviors are described in this section.

When UAC sends the SIP request, UAC can generate nonce before assembling the new authentication header field.

For DAS, UAC must obtain the certificate of DAS for the next-hop node. The nonce is encrypted and inserted into Response-Authentication. For the shared key with the next-hop node, the nonce is encrypted by the shared key to ensure its privacy.

When it receives the SIP response for the corresponding SIP request, UAC should verify the authorization from the next hop. It generates its own digest through its saved nonce in decrypted format, plus some header fields and the message body in response message. This digest is compared with the one in SIP response message from the next hop. If there is a mismatch, it should treat it as an error and may terminate the dialog with the failure reason.

Even if UAC may receive the response code 432 Failed Response Authorization, UAC should finish the steps for verifying the received response from the upstream node. If Response-Authorization carries the correct digest, this response code can be trusted. The proper follow-up operations should take place, such as terminating the dialog with the failure reason. If not, the received response may be suspicious. UAC should analyze the reason before taking any steps for further operations.

As a recipient of the SIP request with Response-Authentication, UAS should generate the digest for SIP response with respect to the specified method. The digest is inserted into UAS's next SIP response message back to the downstream node.

5. Proxy Server Behavior

The extensions in this document require new processing and parsing

for proxy servers. Their behaviors are described in this section.

After receiving the SIP request with Response-Authentication, the proxy server must save the nonce received from the upstream node.

When the proxy server relays the SIP request, it is recommended that the proxy server carry its own Response-Authentication inside the request. The nonce should be encrypted in the specified methods.

Before relaying the SIP request to the next-hop downstream node, the proxy server should generate its own nonce, encrypt the nonce in the specified method, and overwrite Response-Authentication header field inside the SIP request.

For DAS, the nonce is encrypted by the certificate of the next-hop domain and inserted into Response-Authentication. For the shared key with the downstream node, the nonce is encrypted by the shared key to ensure its privacy.

Note the nonce received from the previous hop should not be forwarded to the next hop for reducing the risk of disclosure.

If the SIP response is received, the proxy server must finish two steps. First, it has to verify the authorization from the next-hop downstream node. It generates its own digest through its saved nonce in decrypted format, plus some header fields and the message body in response message. This digest is compared with the one in SIP response message from the next hop.

Second, it has to generate another digest from the decrypted nonce received from the upstream node, some header fields, and the message body for SIP response. This digest is inserted into its relayed SIP response to the upstream node.

Note that the proxy server has to obtain the certificate, the public key or the shared key with the downstream node (or domain) before Response-Authentication is assembled. [4] is recommended to retrieve the certificate through SUBSCRIBE and NOTIFY in the enhanced certificate management.

When it receives the SIP response for the corresponding SIP request, the proxy server should compare the digest inside Response-Authorization with its generated one. If there is a mismatch, the proxy server should analyze this suspicious response. The proper follow-up operations should take place, such as replacing the response code with 432 Failed Response Authorization. Note that the saved digest for the corresponding SIP request should be piggybacked into its response.

Even if it receives the response code 432 Failed Response-Authorization, the proxy server should finish the steps for verifying the validness of this received response from the downstream node.

6. Syntax and Examples

6.1. Header Syntax

Two new SIP headers are introduced in this document. Response-Authorization appear in the response. Response-Authentication is eligible in the request.

```
Response-Authentication = "Response-Authentication"
                          HCOLON resp-authen-param
resp-authen-param = auth-method-param * (SEMI nonce-param)
auth-method-param = "method" EQUAL auth-method-enum
                    * (SEMI alg-param)
auth-method-eum   = "DAS" / "SharedKey" / "PublicKey"
alg-param         = "alg" EQUAL token
nonce-param       = "nonce" EQUAL "nonce-value"

Response-Authorization = "digest" EQUAL resp-author-digest
Resp-author-digest = LDQUOT 32LHEX RDQUOT
```

For the digest generated in Response-Authorization, the digest-string includes

- status code of the response
- addr-spec in To
- addr-spec in From
- addr-spec of claimer field in Responder
- method and nonce in Response-Authentication
- callid from Call-ID
- the digits and the method from CSeq
- Date field
- body content of the message with the bits exactly as they are in the message (in the ABNF for SIP, the message body).

In summary, digest-string for Identity header in the SIP response is

```
digest-string = status-code ":"
                addr-spec ":" addr-spec ":" addr-spec  ":"
                auth-method-enum nonce-value ":"
                callid ":" 1*DIGIT SP method ":" SIP-Date ":"
                message-body
```

The decrypted nonce plus this digest-string is hashed and signed with the key based on the specified method. The mandatory procedure is sha1WithRSAEncryption as described in RFC 3371 with base64 encoding as described in RFC 3548.

One simple example is given below to show how these new header fields are used when Alice sends an INVITE to bob.

```
INVITE sip:bob@biloxi.com SIP/2.0
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Response-Authentication: method=DAS; alg=rsa-sha1;
  nonce=rqupqurcnvajfaqruiopqurewfval4139814kfaj134
        vnnfaq2kqklpijmhyhhbvfdw43ikfr3535wtetwetw
Content-Type: application/sdp
Content-Length: 142
```

The response from Bob should provide Response-Authorization to answer the challenge from Alice.

```
SIP/2.0 200 OK
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Response-Authorization:
       digest=lqkduncbhyr467u8932udjbfsgdiwoopxjnxdg
       wuhfduiiqriqopqr3990mcnvbgdqewzsjdormgbktgui
Content-Type: application/sdp
Content-Length: 131
```

7. Security Considerations

This document provides some complementary security enhancements on SIP response authentication, when TSL is absent on some hops.

For example, if a rogue proxy can sniff the SIP requests from Proxy-1 to Proxy-2 without TLS, it can spoof the addresses and URIs of Proxy-2 and send the response back to Proxy-1 along with its own rogue domain authentication service info, before Proxy-2's response.

Without the proposed mechanisms, Proxy-1 and the initiator of SIP requests will be deceived by the response from the rogue proxy. This allows the rogue proxy to conduct attacks, such as redirecting the requests to attack other targets for DoS attacks, redirecting the requests to rogue users for information disclosure, and terminating the dialogs for turning down SIP services.

With the mechanisms introduced in the document, Proxy-1 can detect the faked responses from the rogue proxy, by checking the digest in Response-Authorization. These faked responses are dropped immediately by Proxy-1 without any impact on the callers of SIP requests.

All the hops with security concerns should apply these mechanisms for enhancing authentication for SIP response, when TLS is absent on those hops. CSRT should be created by combining TLS and this per-hop response authentication. If not, man-in-the-middle attacks may be possible again through SIP response, just as before.

There are some open questions in the future work for enforcing these mechanisms and creating CSRT per SIP dialog. One is how to indicate CSRT is required by the originator UAC. Another is how to notify UAC if CSRT is fully formed or where CRST is missing if applicable.

Another security concern is about nonce used this enhancement. Nonce should be random enough for a long period of time. Nonce during a long SIP session should be refreshed periodically to prevent it from being compromised.

This document is also based on some existing results for domain-based authentication and certificate management (See [3, 4]). Therefore, these mechanisms may be affected by the secure concerns for these functional components.

8. IANA Considerations

This document requests changes to the header and response-code sub-registries of the SIP parameters IANA registry.

8.1. Header Field Names

This document specifies two new SIP headers: Response-Authentication and Response-Authorization. Their syntax is given in Section 6. These headers are defined by the following information, which is to be added to the header sub-registry under http://www.iana.org/assignments/sip-parameters.

Header Name: Response-Authentication
Compact Form: (none)
Header Name: Response-Authorization
Compact Form: (none)

8.2. 432 'Failed Response Authorization Response Code

This document registers a new SIP response code which is described in Section 3.2. It is used when the expected Response-Authorization is missing or doesn't carry the correct digest. This response code is defined by the following information, which is to be added to the method and response-code sub-registry under

http://www.iana.org/assignments/sip-parameters.
Response Code Number: 432
Default Reason Phrase: Bad Identity-Info

9. Contributors' Address

Cullen contributed to the development of this document in every aspect. He helped to define the scope and the essential goals in the beginning. He provided substantial input and rewrote some parts of this documents.

Cullen Jennings
Cisco Systems
170 West Tasman Dr
MS: SJC-21/3

Phone: +1 408 421 9990
EMail: fluffy@cisco.com

10. Acknowledgments

The editor and the contributors would like to acknowledge the constructive feedback and input provided by John Elwell, Jon Peterson, Jonathan Rosenberg, Peter Thermos, Dean Willis, and Rohan Mahy in emails and discussions in IETF meetings.

11. References

11.1. Normative References

[1] 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.

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

[3] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[4] Jennings, C. and J. Peterson, "Certificate Management Service for The Session Initiation Protocol (SIP)", draft-ietf-sipping-certs-02 (work in progress), July 2005.

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

[6] Metz, C., "OTP Extended Responses", RFC 2243, November 1997.

11.2. Informational References

[7] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[8] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[9] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

Author's Address

Feng Cao (editor)
Cisco Systems
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

Email: fcao@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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