SIP Message Manipulation - Setting the P-Asserted Identity

Table of Contents
- Prerequisites
- Summary
- PAI Static Modification
- Configure the SBC Edge
- Debugging Output
- Conditional PAI Modification
- Important Notes

About this Page
- A sneak peek into the SBC Edge's new SIP Header Manipulation Feature. This new feature permits any SIP header to be added, deleted, or modified.
  - This SIP Header Manipulation example uses the Literal manipulation option and shows how perform Header Manipulation only when certain criteria are met.
  - This document uses the Conditional Rules feature to limit Message Manipulation to very specific instances.

Related Articles
- Managing Signaling Groups

Prerequisites
- Requires SBC Edge Version 2.2

Summary
Because you can never tell just what the other side might send or need, the SBC Edge version 2.2 will include a feature to manipulate any SIP header.
- SIP Headers can be added, deleted, or modified.
- Manipulation can be done for every SIP message, or separately for SIP Requests or SIP Responses.
- Manipulations can set to occur based upon user specified matching criteria.
- Manipulations can be executed at the ingress or egress SG. At the ingress, inbound SIP messages are modified to permit proper handling by the SBC's routing function. At the egress, SIP messages already processed by the SBC are modified to meet the SIP requirements of the upstream device.

PAI Static Modification
In today's sneak peek, we modify the PAI at the egress SG for a PSTN > SBC > Lync call. This is an example of a static replacement--a replacement that will take place on every SIP packet that leaves the SBC bound for Lync.

Near the end of the document, we'll add a configuration that will make the PAI modification conditional, only changing the PAI when certain criteria is met.

Configure the SBC Edge
1. Select the new Message Manipulation entry in the navigation tree
2. Add a SIP Message Rule Table

3. Set how the rule will be applied. It can be applied to All SIP Messages, only Requests, only Responses, or Selected Messages.

4. Add an entry to the new table.

5. Set the Header Action to Modify and the Header Name to P-Asserted-Identity. As you can see, any SIP Header can be manipulated.
6. To set the new PAI value, click Add/Edit...

7. ... and add the new value in the Edit Message Field

8. Click OK to complete adding the entry to the table.
9. In the SG, Enable the Message Manipulation

10. Add the new Message Manipulation to the Outbound Message Manipulation Table

Debugging Output

In the debugging output below, you can see the PAI change from +15103644064 to 555

```
[2012-10-03 18:20:32,198] 5397 0001 com.net.ux.sip TRACE (Executive.cpp:1252) - MatchSipMessageRule: StartLine from configuration (*)From Msg (INVITE sip:+14083483531@sba1.vx.net:5067;user=phone SIP/2.0)
[2012-10-03 18:20:32,198] 5400 0001 com.net.ux.sip DEBUG (ProtocolField.h:93) - debugInfo:
```
FieldElement:: FieldType(1) Source(<sip:555@10.1.1.74:5067;user=phone>)
[2012-10-03 18:20:32,199] 5401 0001 com.net.ux.sip DEBUG (ProtocolField.h:211) - debugInfo: 
FieldElementLiteral:: stringLiteral(<sip:555@10.1.1.74:5067;user=phone>)
    HeaderRuleType(1)
[2012-10-03 18:20:32,199] 5403 0001 com.net.ux.sip DEBUG (RuleBase.h:123) - debugInfo: RuleBase::
    RuleBaseType(1) Result(0) Description(test pai) valid(true), Condition Expression(${1})
[2012-10-03 18:20:32,199] 5404 0001 com.net.ux.sip TRACE (Executive.cpp:211) - processMessageRule:
    Processing Header rule
[2012-10-03 18:20:32,199] 5405 0001 com.net.ux.sip DEBUG (Executive.cpp:316) - processHeaderRule:
    Processing header Rule (test pai)
[2012-10-03 18:20:32,199] 5406 0001 com.net.ux.sip TRACE (Executive.cpp:328) - processHeaderRule:
    Processing header Conditions Count=1
[2012-10-03 18:20:32,199] 5407 0001 com.net.ux.sip TRACE (Executive.cpp:640) - processConditionRules:
    evaluateConditionRuleTruth: CREId1
[2012-10-03 18:20:32,199] 5409 0001 com.net.ux.sip DEBUG (Executive.cpp:667) - processHeaderRule:
    Processing header Rule (test pai)
[2012-10-03 18:20:32,199] 5410 0001 com.net.ux.sip TRACE (Executive.cpp:727) - buildField: Building
    fields with element count=1
    field literal
    header modify
    rule Modify ObjectToken:: StringIndex(p-asserted-identity) Ordinal(1)
[2012-10-03 18:20:32,199] 5414 0001 com.net.ux.sip TRACE (Executive.cpp:1311) - getObjectValue:
    name:(3:Header) value:
[2012-10-03 18:20:32,199] 5415 0001 com.net.ux.sip TRACE (Executive.cpp:1391) - setObjectValue:
    name:(3:Header) value:<sip:555@10.1.1.74:5067;user=phone>
[2012-10-03 18:20:32,199] 5416 0001 com.net.ux.sip TRACE (SipMessage.cpp:1311) - getobjectValue:
    name:(3:Header) value:
[2012-10-03 18:20:32,199] 5417 0001 com.net.ux.sip TRACE (SipMessage.cpp:1391) - setObjectValue:
    name:(3:Header) value:<sip:555@10.1.1.74:5067;user=phone>
[2012-10-03 18:20:32,199] 5418 0001 com.net.ux.sip TRACE (Executive.cpp:218) - processMessageRule:
    HeaderRule executed
    matched at least one message rule
    application data size: 990 for conn_id: 427 on handle: 0x2200b8, retry: 0, retry attempts: 0
[2012-10-03 18:20:32,201] 5421 0001 com.net.ux.sip DEBUG (TransportLayer.cpp:538) - 0x293a30 sending
    from [10.1.1.74]:25174 to [10.1.1.73]:5067 990 bytes over 00TLS722-25174 attempt(1) (If:0) ln(1320):
INVITE sip:+14083483531@sba1.vx.net:5067 SIP/2.0
Allow: INVITE, ACK, CANCEL, BYE, UPDATE, NOTIFY, OPTIONS, REFER, REGISTER
Call-ID: e6e51900-0000-0010-0C13-B
Contact: <sip:+15103644064@ux.vx.net:5067;transport=TLS>
Content-Length: 353
Content-Type: application/sdp
CSeq: 1 INVITE
From: <sip:+15103644064@10.1.1.74:5067;user=phone>;tag=a01014a-85
Max-Forwards: 70
P-Asserted-Identity: <sip:555@10.1.1.74:5067;user=phone>
Supported: replaces,update,100rel
To: <sip:+14083483531@sba1.vx.net:5067;user=phone>
User-Agent: NET UX 2.2.0v177 UX
Via: SIP/2.0/TLS 10.1.1.74:5067;branch=z9hG4bK-UX-0a01-014a-02fa

v=0
o=UX 1007 1007 IN IP4 10.1.1.74
s=VoipCall
c=IN IP4 10.1.1.74
t=0 0
m=audio 16432 RTP/SAVP 8 0 101 13
c=IN IP4 10.1.1.74
a=rtpmap:8 PCMA/8000/1
a=rtpmap:0 PCMU/8000/1
a=rtpmap:101 telephone-event/8000
a=rtpmap:13 CN/8000
a=ptime:20
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:oovkXPgyggyXzX9w4gx9s6z1KQnRk/0x67b|2^31|1:1
Conditional PAI Modification

By adding a Conditional Rule to the Message Rule, you can perform the manipulation only when it matches a particular condition. In the example below, the PAI would only be modified if the call is from +15103644064.

1. Add a Conditional Rule Table entry

2. Add the Description and set the Match Type from the pulldown. In this case, we’re looking to match the calling number, which is represented by the from.uri.userinfo.user token.

3. Set the Operation to Equals. Set the Match Value Type to Literal and put the number to be matched into the Match Value.
4. With the Match information configured, it's time to modify the Message Rule so that it only runs when the Condition is met. In the Message Rule, click the **Condition Expression Add/Edit** button.

5. Set it for **Match All Conditions** and select the Conditional Rule you just finished programming in the previous steps. Click **Apply**.

6. Click **Apply** to complete the update to the Message Rule.
### Important Notes

The SIP Header feature cannot **generate** a new SIP packet. Only arriving or leaving SIP packets changed via the SIP Header function.