

SIP Configuration Flexibility

The Session Border Controller (SBC) offers flexibility in configuring the following features of a Session Initiation Protocol (SIP adjacency):

- OPTIONS Support
- Rewriting from header on non-REGISTER requests
- Rewriting To: header on non-REGISTER requests
- Auto-detecting NAT
- Routing on wildcard domains

**Note**

For ACE SBC Release 3.0.00, this feature is supported in the unified model only.

Feature History for SIP Configuration Flexibility

Release	Modification
ACE SBC Release 3.0.00	This feature was introduced on the Cisco7600 series router along with support for the SBC unified model.

Contents

This module contains the following sections:

- [Restrictions for Implementing SIP Configuration Flexibility, page 29-1](#)
- [Information About SIP Configuration Flexibility, page 29-2](#)
- [How to Implement SIP Configuration Flexibility, page 29-3](#)

Restrictions for Implementing SIP Configuration Flexibility

The restrictions for implementing SIP configuration flexibility are listed per feature in this chapter.

Information About SIP Configuration Flexibility

This section contains the following subsections:

- [OPTIONS Support, page 29-2](#)
- [Rewriting From Header on Non-Register Requests, page 29-2](#)

OPTIONS Support

By default, the SBC blocks the OPTIONS method from passing through, but users can now configure the SBC on a per-adjacency basis to pass or block the OPTIONS method. If you configure the SBC with method whitelist profiles per adjacency, then the SBC allows the OPTIONS method to pass through. If you configure the SBC with method blacklist profiles per adjacency, then the SBC blocks the OPTIONS method from being passed through.

Restrictions for OPTIONS Support

- The SBC strips out SDP blocks from messages when it allows the OPTIONS method to pass through. This limits what the SIP endpoints can exchange.
- The SBC-SIG does not send the Accept and Allow headers on any methods, including OPTIONS.
- The SBC allows only the 100Rel and Replaces tags of the Supported header to pass through, while the other tags of this header are controlled by whitelists and blacklists.

Rewriting From Header on Non-Register Requests

With this feature, users can configure the SBC on a per-adjacency basis to control whether it rewrites the hostport section of the From header on Non-Register Requests to the outbound SIP adjacency address or port. If the SBC is configured to allow the From header to pass through without it being rewritten, then the SBC allows the entire header to pass through without changing it. The only exception occurs with the Tag parameter; the SBC assigns a different value to this parameter before passing it through.

Restrictions for Rewriting From Header on Non-REGISTER Requests

- This feature is not applicable for REGISTER requests.
- This feature may only work limitedly with the Rewrite-Register feature.
- If the From header contains a Tel URI, then the SBC does not rewrite the header since it does not have a hostport.
- Depending on the number of headers, options and SIP whitelist profiles, the SBC limits the size of the From header that it allows to pass through to approximately 1000 bytes.

Rewriting To: Header on Non-REGISTER Requests

With this feature, users can configure the SBC on a per-adjacency basis to control whether a SIP To: header in SIP requests is rewritten by the SBC or not. Current behavior (and the default) is to always rewrite the hostport in the To: header to the outgoing adjacency local id/address/port and to strip any parameters. New configurable behavior will be to pass through the To: header unchanged.

If the adjacency option is configured to rewrite To: header, any dialog-creating or out-of-dialog SIP requests sent from this adjacency, except for REGISTER requests, will have the To: header rewritten to match the outgoing Request-URI. If disabled, the To: header will be passed through unchanged, including any parameters. This option has no effect on in-dialog requests, which always use the To: header established at dialog creation.

Auto-detecting NAT

With the addition of a new configuration field to the SIP adjacency, it is now possible for users to specify if the SBC must auto-detect whether a NAT is in use on that adjacency. If the SBC is configured to auto-detect NAT, then for each request that it receives, the SBC determines whether a NAT is in use for that endpoint. If the SBC determines that NAT is in use, then the SBC stores the bindings for that request and uses them when sending a response. Additionally, the SBC stores and reuses bindings for REGISTER requests for subsequent Dialog-forming and Out-of-dialog requests.

Restrictions for Auto-detecting NAT

- The SBC can auto-detect NAT only by comparing the Sent-by stopper in the Via header with the remote address and port of the message.
- If the stopper contains a domain name, instead of an IP address, the SBC cannot auto-detect whether NAT is in use. In this case, the SBC assumes that NAT is in use.
- Auto-detecting NAT is applied only to Out-of-dialog requests or Dialog-forming requests.

Routing on Wildcard Domains

The SBC routing policy allows you to use the * character in a text domain name match string. This character can match any number of characters in the called address. For example, *domain.com can match both sip1.domain.com and sip2.domain.com.

Restrictions for Routing on Wildcard Domains

- You can only specify one wildcard character in a given match string.
- This feature applies only to text domain name match rules, and not to dialed digit match rules.

How to Implement SIP Configuration Flexibility

This section contains the steps for implementing SIP configuration flexibility.

SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **passthrough from header**

6. `passthrough to header`
7. `nat force-on`
8. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure</code> Example: host1/Admin# <code>configure</code>	Enables global configuration mode.
Step 2	<code>sbc service-name</code> Example: host1/Admin(config)# <code>sbc mysbc</code>	Enters the mode of an SBC service. <ul style="list-style-type: none"> • Use the <i>service-name</i> argument to define the name of the service.
Step 3	<code>sbe</code> Example: host1/Admin(config-sbc)# <code>sbe</code>	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	<code>adjacency sip adjacency-name</code> Example: host1/Admin(config-sbc-sbe)# <code>adjacency sip sipadj</code>	Enters the mode of an SBE SIP adjacency. <ul style="list-style-type: none"> • Use the <i>adjacency-name</i> argument to define the name of the SIP adjacency.
Step 5	<code>passthrough from header</code> Example: host1/Admin(config-sbc-sbe-sip-adj)# <code>passthrough from header</code>	Configures the SIP adjacency to disable From rewrite.
Step 6	<code>passthrough to header</code> Example: host1/Admin(config-sbc-sbe-sip-adj)# <code>passthrough to header</code>	Configures the SIP adjacency to disable To rewrite.
Step 7	<code>nat force-on</code> Example: host1/Admin(config-sbc-sbe-sip-adj)# <code>nat force-on</code>	Configures the SIP adjacency to assume that all endpoints are behind a NAT device. To configure the SIP adjacency to assume that no endpoints are behind a NAT device, use the nat force-off command. By default, the SBC autodetects whether the endpoints are behind a NAT device.
Step 8	<code>exit</code> Example: host1/Admin(config-sbc-sbe-sip-adj)# <code>exit</code>	Exits the adj-sip mode and returns to the SBE mode.