

## SIP Outbound Authentication

---

The Session Border Controller (SBC) supports Session Initiation Protocol (SIP) outbound authentication. When network entities communicate using SIP, one entity often needs to challenge another one to determine if it is authorized to transmit SIP signaling into the challenger's network. The SIP authentication model is based on the HTTP digest authentication, as described in the RFC 2617.



**Note**

The use of basic authentication, where passwords are transmitted unencrypted, is not permitted in SIP.

---



**Note**

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

---

### Feature History for SIP Outbound Authentication

| Release                | Modification  |
|------------------------|---|
| ACE SBC Release 3.0.00 | This feature was introduced on the Cisco 7600 series router along with support for the SBC unified model. |

## Contents

This module contains the following sections:

- [Prerequisites for Implementing SIP Outbound Authentication, page 26-2](#)
- [Restrictions for Implementing SIP Outbound Authentication, page 26-2](#)
- [Information About SIP Outbound Authentication, page 26-2](#)
- [How to Configure SIP Outbound Authentication, page 26-3](#)
- [Examples of Show Commands, page 26-5](#)

# Prerequisites for Implementing SIP Outbound Authentication

The following prerequisites are required to implement SIP outbound authentication:

- Configure a SIP adjacency before you specify one or more authentication-realms.
- Configure the SBC with a set of domains (realms) with which it can authenticate itself. Set the username and password to provide when challenged by each of these domains. This configuration is implemented per adjacency.

**Note**

---

Multiple realms can be configured per adjacency and there is no limit on the number of these realms aside from memory availability. Different realms may be configured with the same username and password. Also, each realm may be configured with different username and password on different adjacencies. However, any realm can be configured a maximum of one time per adjacency.

---

# Restrictions for Implementing SIP Outbound Authentication

The following restrictions apply to SIP outbound authentication:

- The SBC rejects any attempt to configure an authentication-realm with the same domain name as an existing authentication-realm. This restriction is valid per adjacency. Multiple adjacencies may have authentication-realms configured with the same domain.

**Note**

---

The current command line interface (CLI) prohibits the user from configuring two authentication-realms with the same domain for the same adjacency. If this is attempted, the CLI interprets the second authentication-realm configuration as an attempt to reconfigure the first authentication-realm, and updates the user's credentials accordingly.

---

- Each authentication-realm can only be configured with a single username and password per adjacency.

# Information About SIP Outbound Authentication

This section contains the following subsections:

- [Configuring Outbound Authentication in the SBC, page 26-2](#)
- [Authenticating the SBC to Remote Devices, page 26-3](#)

# Configuring Outbound Authentication in the SBC

When a SIP adjacency is configured, the user may specify one or more authentication-realms. Each authentication-realm represents a remote domain, from which the SBC receives authentication challenges on the adjacency. When an authentication-realm is configured, the user must specify the correct user name and password that the SBC uses to authenticate itself in that realm. The SBC stores all valid authentication-realms for each adjacency.

## Authenticating the SBC to Remote Devices

Upon receipt of a SIP 401 or 407 response that can be correlated to a request it sent, the SBC examines the attached authentication challenge. The SBC responds to any authentication challenge received on a given adjacency that matches one of the configured authentication-realms for that adjacency. Any authentication challenge that does not match the configured authentication-realm is passed through unchanged to the SBC's signaling peer for the adjacency, on which the original request was received.

To generate a response to an authentication challenge, the SBC does the following:

1. First, it looks up the realm parameter of the challenge in its list of configured authentication-realms for the outbound adjacency.
2. Second, it finds the password for that authentication-realm and generates an authentication response by combining the password with the nonce parameter from the challenge, and hashing the result.
3. If the challenger has requested **auth-int** quality of protection, the SBC also generates a hash of the entire message body and includes it in the response.
4. The SBC builds an Authorization (or Proxy-Authorization) header by including the following parameter values (following RFC 2617):
  - Nonce from challenge.
  - Realm from challenge.
  - Digest-URI is set to the SIP URI of the challenged request.
  - Message-QOP is set to **auth**.
  - Response calculated as described previously.
  - Username as specified for the relevant authentication-realm.
  - If the challenge contained an **opaque** parameter, it is returned unchanged on the response.
  - If the challenge contained the **qop-directive** parameter, then the **nonce-count** parameter is set to the number of the sent requests, using the response calculated from this nonce.
  - Note that the domain parameter is not expected to be included on any challenges that the SBC must respond to. This parameter is not used on Proxy-Authenticate challenges, the type of challenge that the SBC most often receives. If the domain parameter is included, the SBC ignores it.
5. Finally, the SBC stores its calculated response and the received nonce with the other data for the authentication-realm. This allows the SBC to respond rapidly to the subsequent challenges from this realm with the same nonce. If the SBC lacks the resources to store its response, it carries on anyway. The next time an authorization challenge is received from this realm, the SBC has to recalculate its response. When the SBC re-uses a saved response, it updates the nonce count stored along with the nonce-response pair. This allows the SBC to correctly fill in the **nonce-count** field in Authorization responses.

## How to Configure SIP Outbound Authentication

This section contains the steps for configuring SIP outbound authentication, allowing the user to add/remove one or more authentication-realms to/from an adjacency.

### SUMMARY STEPS

1. **configure**

2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **authentication-realm inbound** *domain* | **outbound** *domain username password*
6. **exit**
7. **show services sbc** *sbc-name* **sbe adjacency** *adjacency-name* **authentication-realms**
8. **show services sbc** *service-name* **sbe all-authentication-realms**

## DETAILED STEPS

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 1 | <b>configure</b><br><br><b>Example:</b><br>host1/Admin# configure  | Enables global configuration mode.   |
| Step 2 | <b>sbc</b> <i>service-name</i><br><br><b>Example:</b><br>host1/Admin(config)# sbc mysbc  | Enters the mode of an SBC service. <ul style="list-style-type: none"> <li>• Use the <i>service-name</i> argument to define the name of the service.</li> </ul>   |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>host1/Admin(config-sbc)# sbe  | Enters the mode of the signaling border element (SBE) function of the SBC.   |
| Step 4 | <b>adjacency sip</b> <i>adjacency-name</i><br><br><b>Example:</b><br>host1/Admin(config-sbc-sbe)# adjacency sip test   | Enters the mode of an SBE SIP adjacency. <ul style="list-style-type: none"> <li>• Use the <i>adjacency-name</i> argument to define the name of the service.</li> </ul>   |
| Step 5 | <b>authentication-realm inbound</b> <i>domain</i> <b>outbound</b> <i>domain username password</i><br><br><b>Example:</b><br>host1/Admin(config-sbc-sbe-adj-sip)#<br>authentication-realm example.com usersbc<br>passwrdsbc | Configures a set of authentication credentials for the specified domain on the specified adjacency. This command can be issued either before or after the adjacency has been attached.<br><br>The <b>no</b> version of this command deconfigures the authentication-realm on the specified adjacency. <ul style="list-style-type: none"> <li>• <b>inbound</b>—Specifies inbound authentication realm.</li> <li>• <b>outbound</b>—Specifies outbound authentication realm.</li> <li>• <b>domain</b>—Name of the domain for which the authentication credentials are valid.</li> <li>• <b>username</b>—User name that identifies the SBC in the specified domain.</li> <li>• <b>password</b>—Password to authenticate the username in the specified domain.</li> </ul> |

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 6 | <b>exit</b><br><br><b>Example:</b><br>host1/Admin(config-sbc-sbe-adj-sip)# exit  | Exits the adj-sip mode and returns to the SBE mode.                                   |
| Step 7 | <b>show services sbc <i>sbc-name</i> sbe adjacency <i>adjacency-name</i> authentication-realms</b><br><br><b>Example:</b><br>host1/Admin# show services sbc mySbc sbe adjacency SipToIsp42 authentication-realms | Shows all currently configured authentication-realms for the specified SIP adjacency. |
| Step 8 | <b>show services sbc <i>service-name</i> sbe all-authentication-realms</b><br><br><b>Example:</b><br>host1/Admin# show services sbc mySbc sbe all-authentication-realms  | Shows all currently configured authentication-realms for all SIP adjacencies.         |

## Examples of Show Commands

```
# show services sbc mySbc sbe adjacency SipToIsp42 authentication-realms
Configured authentication realms
-----
Domain      Username  Password
Example.com usersbc   passwordsbc
```

```
# show services sbc mySbc sbe all-authentication-realms
Configured authentication realms
-----
Adjacency: SipToIsp42
Domain      Username  Password      Example.com  usersbc      passwordsbc
Remote.com  usersbc   sbcpassword
```

```
Adjacency: SipToIsp50
Domain      Username  Password
Example.com user2sbc  password2sbc
Other.com   sbcuser  sbcsbcsbc
```

