Overview

The SIPREC protocol defines the interaction between a Session Recording Client (SRC) and a Session Recording Server (SRS), and controls the recording of media transmitted in the context of a communications session (CS) between multiple user agents. The Recording Session (RS) is established over SIP from SRC to SRS.

The SBC Core supports SIPREC towards multiple recorders based on the Internet Engineering Task Force (IETF) standard. The SBC acts as an SRC sending recording sessions to a third-party SRS when a configured recording-criteria is met.

Session recording is used for various purposes such as complying with regulation, monitoring quality of service of representatives as well as storing call information for quality analysis.

The SBC Core supports the following proprietary SIP recording interfaces:

- SIPREC SIP-based session recording
- Call monitoring MCT
- NICE session recording

Access to the Media Capture Tool is restricted to privileged, password-protected user accounts. Tracking of its use is tracked by AUD logging.

Note
The SBC can record all the calls in the system. The number of recording sessions depends on the available interface bandwidth.

Table 1: Supported RFCs

<table>
<thead>
<tr>
<th>Category</th>
<th>RFCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Cases and Requirements for SIP-Based Media Recording (SIPREC)</td>
<td>RFC 6341</td>
</tr>
<tr>
<td>Media Recording Architecture</td>
<td>RFC 7245</td>
</tr>
<tr>
<td>Session Initiation Protocol (SIP) Recording Metadata</td>
<td>R0FC 7865</td>
</tr>
<tr>
<td>Session Recording Protocol</td>
<td>RFC 7866</td>
</tr>
<tr>
<td>Session Initiation Protocol (SIP) Recording Call Flows</td>
<td>RFC 8068</td>
</tr>
</tbody>
</table>
In the figure SBC SIP Recording Strategy, the basic call is established between SIP phone 1 and SIP phone 2 through the SBC, which is known as communication session (CS). The SBC establishes an RS based on CS towards SRS. The SBC and SRS may exist in the same or different administrative domains.

**Figure 1: SBC SIP Recording Strategy**

Configuring SIPREC Based Recording

The two methods to trigger a call recording are:

- A call matches call recording criteria causing the Policy Server to trigger the SBC to record the call.
- Initiate recording via CLI using GCID.

PSX Configuration

The recording-criteria determine, which sessions to record, SRS information along with other operational options.

The PSX/ERE uses the following configurable objects when determining whether a call needs to be recorded or not:

- Recording Criteria—contain the rules to match for invoking call recording (this is same for SIPREC and MCT).
- Recorder Profile—contains information about the recorder to use for a particular Recording Entity (obsolete—used in older versions when SRS redundancy is not supported).
  - Contains
    - Transport
    - IP V4/V6 address and port
- SRS Groups—contains multiple Recording profiles for SRS redundancy (up to 8).
  - Contains data of multiple SRS servers
    - Transport
    - IP V4/V6 address port
    - Encryption data (for SRTP)
    - IP TG to be used by the SBC for RS session
- Recording Cluster profile—contains multiple SRS Groups for simultaneous recording (up to 4).

**Note**
The PSX/ERE supports provisioning 128 Recording criteria, 256 SRS Group Profiles and 256 SRS Cluster Profiles.

Recording Criteria
The need to record a call is decided from the PSX based on the following criteria in the given order of priority:

- Recorder type
  - SIPREC
  - MCT
- Next hop signaling IP address
- Previous hop signaling IP address
- Calling Party Number
- Called Party number
- Ingress TG ID
- Egress TG ID
- GATEWAY

For detailed configuration information, refer to the section Deploying SBC For SIPREC

**CLI Triggers**

- Recording RTP session can be initiated for a call by providing its GCID through CLI. The user provides the IP Address/port for the corresponding session recorder through the same CLI command.
- Listing the calls currently being recorded using a CLI command. The SBC displays their GCID, and the RTP destinations and the SRS IP address.

To enable/disable SIPREC feature, use following syntax:

```
% set addressContext <ADDRESS-CONTEXT> zone <ZONE> sipSigPort <SIP SIGNALLING PORT> siprec <disabled|enabled>
```

To start/stop a recording, the following CLI syntax applies:

```
% request global sipRec startRecord gcid <GCID> callLeg ingress numOfStreams <Number of recorders 1 or 2> srsIpAddress <SRS IP ADDRESS> srsPort <SRS PORT> transport <tcp | udp> trunkGroup <TRUNK GROUP NAME> srsIpAddress2 <SRS IP ADDRESS> srsPort2 <SRS Port> transport2 <tcp | udp> trunkGroup2 <SIP Trunk Group>
% request global sipRec stopRecord gcid <GCID> recorderAddress <IP Address> recorderPort <Port Number>
```

**Note**

If only the GCID value is mentioned in the stopRecord, all the multiple recordings for that GCID are stopped at once.

To view SIPREC status, use CLI syntax:

```
> show table global SipRecStatus
  RECORDER  RX RTP  TX RTP  RECORDING
  GCID  ADDRESS  ADDRESS  ADDRESS  LEG
```

Refer to Zone - SIP Sig Port - CLI and Request Global - CLI pages for CLI command details.

**Recording Procedure**

Once the SBC determines that the call must be recorded, it initiates the SIP INVITE towards the SRS specified in the recording criteria.

- Includes a "+sip.src" feature tag extension in the Contact URI
- Options Tag "siprec" in INVITE towards SRS.
- Two m= lines one for the Rx stream and one for the Tx stream.
  - The attribute "a=label" identifies the streams in the metadata.
**Note**
The SBC does not support transcoding towards the SRS. If the SRS replies back with any other codec, the recording session continues until the SRS terminates the call on its own.

**Note**
- The SBC does not support “Recording Aware UEs”
  - If an INVITE is received from UE with a options tag “require: siprec”, the SBC rejects the request with a 4xx message.
- The SBC does not support SRS initiated recording
  - If an INVITE is received from an SRS with a options tag “require: siprec”, the SBC rejects the request with a 4xx message.
  - If any request except session keepalive re-INVITE/UPDATE or BYE is received in the context of an RS, the SBC rejects the request with a 405 "Method Not Allowed" message.

For configuring SIPREC feature, refer to the section [Deploying SBC For SIPREC](#).

**Note**
The SBC stops recording a call in one of the three ways:
- Providing GCID through CLI recording STOP command
- Through normal CS Sessions disconnect
- Receiving BYE from the SRS

**Supported Features**

The SBC supports following SIPREC features:
- The SIPREC is supported on UDP, TCP, and TLS protocols.
- The SBC supports recording SIP and SIP-I calls.
- If Options PING mechanism is configured on the IP peer, it is used as a keep-alive mechanism for all the SRSs.
- Troubleshooting a SIPREC recorded call using MCT is only possible when initiated through CLI.
- SRS Redundancy
- Simultaneous recording
- Dynamically programmable metadata content
- The SBC records the SIPREC recording information in the CDR.

**Note**
The SIPREC feature is controlled by a system-wide SBC license (SBC-SIPREC). If the license is not available, any SIPREC recording returned by a PSX is ignored.

**SRS Redundancy**

The SBC supports the concept of Primary and secondary SRS servers for redundancy. It supports multiple (up to 8) SRS servers in a SRS Group. All of them can be active at any point of time.
- The SBC can load balance among the recorders in the SRS Group in sequential or round-robin fashion.
- The SBC also supports crank-back mechanism where it tries the next SRS if an SRS does not respond to the RS INVITE or rejects the
RS INVITE with a 4xx.

- The SBC also maintains the health check of the SRS servers using Options PING. If an SRS is marked not reachable, the SBC automatically uses the next server in the SRS group for the next RS session.

Note
SRS Redundancy is supported only when numOfStreams is set to “1” in an SRS Group. When numOfStreams = “2”, the SBC is already sending media to the redundant recorder.

Note
The SRS is not placed into a blacklist if it fails to respond to a RS INVITE because the blacklist is updated only by the path-check mechanism upon failure to respond to OPTIONS Ping.

Simultaneous Session Recording - SIP Ingress and Egress Legs

The SBC is enhanced to support simultaneously recording SIP egress and ingress legs during a session, for a total of four recordings (four simultaneous streams: two in the ingress leg, and two in the egress leg).

The SBC provisions the SIPR recordings towards all 4 recorders, two from Ingress tap point and another two from egress tap point. (Due to NP limitations, four simultaneous recordings cannot be triggered on the same call leg.)

Note
- The SBC supports support sending the recording streams to up to four SRS servers simultaneously.
- Each recording criteria can be configured with a Recording Cluster. A Recording Cluster can have up to four SRS Groups.

Note
- If there is more than one SRS associated with the SRS Cluster, the SBC records on both the legs (Ingress and Egress). The first two recordings are on Ingress leg and the rest on Egress.
- For Quad SIPREC, there are four recordings triggered. Two recordings are triggered on the Ingress leg and two on the Egress leg.
- If there is more than one SRS Group configured, it is recommended to set recordingType to “both legs” or “all legs”.

Example of Simultaneous Session Recording

Below is a diagram illustrating the use case of a simultaneous SIPREC, with first two recordings on Ingress call-leg, and the next two recordings on egress call-leg.

Figure 2: An Example of Simultaneous Session Recording
For more information on parameter configurations and CDR field descriptions refer to:

CLI and EMA:

- Request Global - CLI
- Servers - SRS Group Profile - CLI
- Servers - Call Recording Criteria - CLI
- Servers - SRS Group Profile - CLI
- Zone - SIP Sig Port - CLI
- Global - SIRpec
- Global - Servers - Srs Group Profile

Alarms:

- sonusSbxSipRecSrsSelectionFailedNotification - MINOR
- sonusSbxSIPRecLicenseNotAvailable - MINOR

CDR:

- CDR Field Descriptions

**SRTP Support for SIPREC Towards SRS**

The SBC supports sending encrypted media streams (Secure Real-Time Transport Protocol (SRTP)) towards the SIPREC recorders.

- The SBC sends the SRTP streams as received from the endpoints.
- The SBC is configured to perform a different encryption (using dedicated crypto suite profile) towards the Session Recording Server (SRS).
- The SBC is configured to send plain RTP packets even when the original Communication Session (CS) is an SRTP call.

With this feature, the SBC:

- Supports SRTP (encrypted streams) towards SIPREC server.
- Provides SIPREC functionality with SRTP calls.
- Provides flexibility by supporting both SRTP or RTP towards the SRS based on the input control in the srsGroupData.
- Supports sending SRTP packets (encrypted media streams) as received from the endpoints (relay/pass-through).
- Supports sending SRTP packets with different encryption techniques using new crypto suite profile configured in the srsGroupData.
• Provides support to terminate original SRTP call at the SBC for the SIPREC, such that:
  • The encrypted media packets are decrypted at the SBC and are re-encrypted again towards the user endpoint using the
    negotiated crypto suite profile (CS) between the SBC and the UE.
  • The decrypted media packets towards SRS can be:
    • Encrypted using the same CS negotiated crypto suite profile.
    • Encrypted using different crypto suite profile as provided by the srsGroupData.
    • Send decrypted media packets (RTP) only towards SRS based on the configuration (CS crypto is ignored).

The following two options are added to the srsGroupData:

• srt: Specifies whether SRTP is enabled for the SRS or not.
• cryptoSuiteProfile: If SRTP is enabled, encrypt recording session using this crypto details.

Note
  • If SRTP is disabled towards SRS, the SBC terminates original SRTP call as it is supposed to decrypt the media towards SRS.
  • If SRTP is enabled and provides its own crypto suite profile, the SBC terminates original SRTP call to encrypt using the srsGr
    oupData crypto suite profile.

When SRTP is disabled for the SRS, the SBC sends unencrypted streams towards the SRS irrespective of the CS is using RTP or SRTP.

When SRTP is enabled for the SRS and if the CS leg that is recorded is not using SRTP:

• If cryptoSuiteProfile is configured for the SRS, the SBC sends the SRTP packets using the cryptoSuiteProfile on the
  recorded leg towards the SRS.

• If cryptoSuiteProfile is not configured for the SRS, the SBC sends the RTP packets.

When the CS is using SRTP pass-through:

• If the call is recorded and SRTP is enabled for the SRS, the SBC relays the SRTP packets as received from the user endpoint.
• If SRTP is enabled for the SRS and cryptoSuiteProfile is configured, the SBC re-encrypts the media using the configured cryptoS
  uiteProfile.

When the CS is an SRTP terminated call:

• If cryptoSuiteProfile is configured for the SRS, the SBC sends the SRTP packets using the cryptoSuiteProfile on the
  recorded leg towards the SRS.
• If cryptoSuiteProfile is not configured for the SRS, the SBC uses the cryptoSuiteProfile from the CS and sends SRTP
  packets.

Note
The SIPREC functionality fails and the alarm sonusSbxSipRecSrsSelectionFailedNotification is generated in case of following
scenarios:

• When original CS call is SRTP and the SIPREC is triggered through CL.
• When DTLS-SRTP call is supported.
• When SBC offers SRTP towards SRS and the SRS answers with either invalid crypto or with RTP. In this scenario, the SBC
  tries to find an "offer-answer" match with next available SRS (SRS redundancy) before failing the SIPREC.

Dynamically Programmable Metadata Content

The SBC supports SIPREC when the SIPREC specifications were in early drafts (draft-ietf-siprec-xx-06). With the implementation of this feature,
the SIPREC standard has evolved to RFCs (RFC 7245, RFC 7865, RFC 7866, and RFC 8068), and provides capability for supporting
"dynamically programmable" selection of metadata content.

• The profile sipRecMetaDataProfile is introduced to the services to provide the capability to configure the headers that are mapped
  from the target call leg to the XML and the corresponding metadata XML element name.
• In case of a basic call, all information is copied from the initial-INVITE message on the leg where the tap is, to the metadata XML.
  However, "To" header and "to-tag" is copied additionally from the local information (as to-tag does not present in the INVITE).
• In case of SIPREC trigger during REFER based transfer, irrespective of where the SIPREC tap is, all information is copied from the
initial-INVITE of the new call leg towards the transfer target (C party).

- In case of CLI triggered recording, the existing implementation of sending predefined information in metadata XML remains same (gcid, call-id, from, to). The new configuration of header-metadata mapping is not considered in this scenario.

The following call flow diagram displays the XML tag name.

**Figure 3: Call Flow**

![Call Flow Diagram]
Example SIP INVITE

An example SIP INVITE is shown below:

INVITE sip:SIPREC-SRS@10.54.80.8:51802 SIP/2.0
Via: SIP/2.0/UDP 10.34.171.39:5060;branch=z9hG4bK00B00021f4cdc2590c2
From: "SIPREC-SRC" <sip:SIPREC-SRC@10.34.171.39>;tag=gK00000237
To: "SIPREC-SRS" <sip:SIPREC-SRS@10.54.80.8>
Call-ID: 35651585_16777218_133945398@10.34.171.39
CSeq: 787532 INVITE
Max-Forwards: 70
Allow:
INVITE, ACK, CANCEL, BYE, REGISTER, REFER, INFO, SUBSCRIBE, NOTIFY, PRACK, UPDATE, OPTIONS, MESSAGE, PUBLISH
Accept: application/sdp, application/rs-metadata-request, application/rs-metadata
Contact: "SIPREC-SRC" <sip:SIPREC-SRC@10.34.171.39>; +sip.src
Require: siprec
Supported: timer, 100rel
Session-Expires: 1800
Min-SE: 90
Content-Length: 4560
Content-Type: multipart/mixed; boundary=sonus-content-delim
MIME-Version: 1.0
--sonus-content-delim
Content-Disposition: session; handling=required
Content-Length: 296
Content-Type: application/sdp

v=0
o=Sonus_UAC 748003 60371 IN IP4 10.34.171.39
s=SIP Media Capabilities
t=0 0
m=audio 1052 RTP/SAVP 0
c=IN IP4 10.54.4.51
a=label:1
a=crypto: AES_CM_128_HMAC_SHA1_80 inline:k3NkQB3Tkr23twOMMjd8YjvLI/XPdgE+a1D8FDho
a=rtpmap:0 PCMU/8000
a=sendonly
a=maxptime:10
m=audio 1050 RTP/SAVP 0
c=IN IP4 10.54.4.51
a=label:2
a=crypto: AES_CM_128_HMAC_SHA1_80 inline:k3NkQB3Tkr23twOMMjd8YjvLI/XPdgE+a1D8FDho
a=rtpmap:0 PCMU/8000
a=sendonly
a=maxptime:10

--sonus-content-delim
Content-Disposition: recording-session
Content-Length: 3976
Content-Type: application/rs-metadata+xml

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording:1'>
  <datamode>complete</datamode>
  <group group_id="OTIxYzk4MDAtN2RkYy0xMA==">
    <associate-time>2018-08-09T08:31:44Z</associate-time>
  </group>
</recording>
<callData xmlns='http://ribboncommunications.com/siprec/calldata'>
  <xTo>&lt;sip:+1999@10.54.80.8:51801;user=phone&gt;;tag=1</xTo>
  <xVia>SIP/2.0/UDP 10.34.171.34:5060;branch=z9hG4bK00B0000a25afeb7eee5</xVia>
  <xSeq>84797 INVITE</xSeq>
  <xFrom>"sipp" <sip:sanrayana@10.34.171.34>;tag=gK0000011e</xFrom>
  <xContentType>application/sdp</xContentType>
  <xMaxForwards>70</xMaxForwards>
  <srsgrpId>GR1</srsgrpId>
  
  <xAcceptContact>*;+g.3gpp.icsi-ref="urn%3Aurn-%7%3A3gpp-service.ims.icsi.mmtel"</xAcceptContact>
  
  <xPPreferredIdentity>"sipp"
  <sip:sanrayana@10.54.80.8:5061"></xPPreferredIdentity>
  <mprofileVers>v1.0</mprofileVers>
  <gcid>356515858</gcid>
</callData>

</group>

<session session_id="OTIxYzlhM2UtN2RkYy0xMA==">
  <group-ref>OTIxYzk4MDAtN2RkYy0xMA==</group-ref>
  <start-time>2018-08-09T08:31:44Z</start-time>
</session>

<participant participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <nameID aor="sanrayana@10.34.171.34:5060">
    <name xml:lang="en">sipp</name>
  </nameID>
</participant>

<participant participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <nameID aor="+1999@10.54.80.8">
    <name xml:lang="en"></name>
  </nameID>
</participant>

<stream stream_id="OTIxYzk4MDQtN2RkYy0xMA==">
  <label>1</label>
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</stream>

<stream stream_id="OTIxYzk4MDUtN2RkYy0xMA==">
  <label>2</label>
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</stream>

<sessionrecordingassoc session_id="OTIxYzlhM2UtN2RkYy0xMA==">
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</sessionrecordingassoc>

<participantsessionassoc participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</participantsessionassoc>

<participantsessionassoc participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</participantsessionassoc>

<participantsessionassoc participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <associate-time>2018-08-09T08:31:44Z</associate-time>
</participantsessionassoc>

<participantstreamassoc participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <send>OTIxYzk4MDQtN2RkYy0xMA==</send>
  <recv>OTIxYzk4MDUtN2RkYy0xMA==</recv>
</participantstreamassoc>

<participantstreamassoc participant_id="OTIxYzk4MDITnN2RkYy0xMA==">
  <send>OTIxYzk4MDUtN2RkYy0xMA==</send>
</participantstreamassoc>