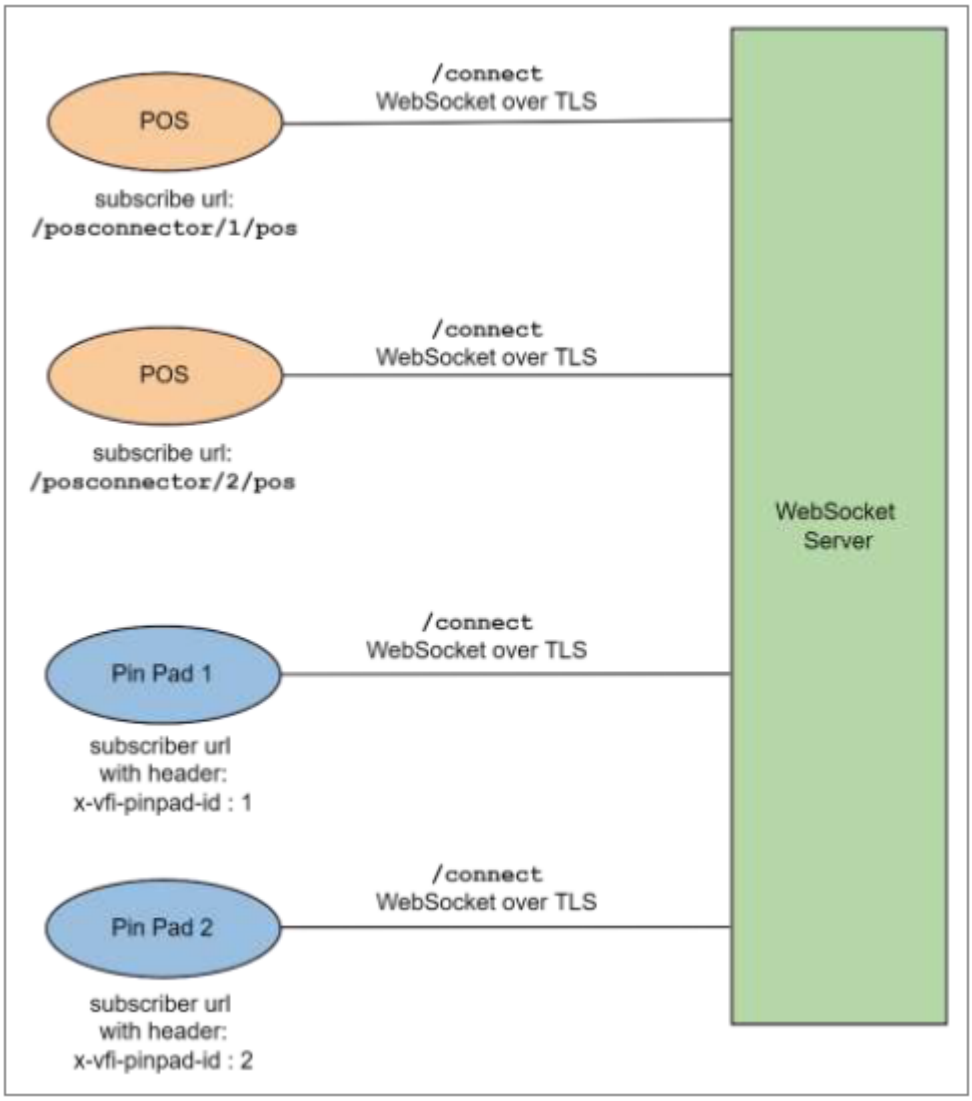
https://verifone.cloud/docs/sca-functional-specification/getting_started/websocket_config
Updated: 20-May-2025

WebSocket

WebSocket is designed to work over HTTP ports 80 and 443 as well as to support HTTP proxies and intermediaries. For example, WebSocket applications can use the standard HTTP ports 80 and 443, thus allowing the use of existing firewall rules. The WebSocket Protocol has two parts: a handshake to establish the upgraded connection and then the actual data transfer. First, a client requests a WebSocket connection by using the Upgrade: WebSocket
and Connection: Upgrade headers, along with a few protocol-specific headers to establish the version being used and set up a handshake. The Upgrade header field is an HTTP header field introduced in HTTP/1.1. In the exchange, the client begins by making a cleartext request, which is later upgraded to a newer HTTP protocol version or switched to a different protocol.
The server, if it supports the protocol, replies with the same Upgrade: WebSocket and Connection: Upgrade headers and completes the handshake. Once the handshake is completed successfully, data transfer begins. Using WebSocket, either the client or the server can initiate communication after the connection is established.

- WebSocket(ws) Protocol is built on top of http.
- WebSocket Secure(wss) Protocol is built on top of https



Example Request/Response

Following is a sample for connection Request sent from terminal to WebSocket POS server:

```
GET / HTTP/1.1\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
Host: 192.168.29.207\r\n
Origin: http://192.168.29.207\r\n
Upgrade: websocket\r\n
Connection: Upgrade\r\n
Sec-WebSocket-Key: CujtCKpQPzowD1oV4qp+hQ==\r\n
Sec-WebSocket-Protocol: sci\r\n
Sec-WebSocket-Version: 13\r\n
x-vfi-pinpad-id : 1\r\n\r\n
```

Following is a sample for connection Response sent from WebSocket POS server to terminal:

```
HTTP/1.1 101 Switching Protocols\r\n
Upgrade: websocket\r\n
Connection: Upgrade\r\n
Sec-WebSocket-Accept: Y2nXMyfwdnd360ikQgJzX7r3Gak=\r\n
Date: Tue, 12 Apr 2022 07:14:07 GMT\r\n
Server: Python/3.10 websockets/10.2\r\n\r\n
```

Once connection is established, SCA will keep listening on this connection for any POS request.
The POS Request will be sent by POS WebSocket to SCA without any headers.

```
<TRANSACTION>
    <FUNCTION_TYPE>SESSION</FUNCTION_TYPE>
    <COMMAND>START</COMMAND>
    <BUSINESSDATE>20210709</BUSINESSDATE>
    <SWIPE_AHEAD>1</SWIPE_AHEAD>
    <TRAINING_MODE>0</TRAINING_MODE>
    <INVOICE>123456</INVOICE>
</TRANSACTION>
```

The POS Response will be sent back without any header by SCA to POS WebSocket.

```
<RESPONSE>
    <RESPONSE_TEXT>Session Started</RESPONSE_TEXT>
    <RESULT>OK</RESULT>
    <RESULT_CODE>-1</RESULT_CODE>
    <TERMINATION_STATUS>SUCCESS</TERMINATION_STATUS>
    <TRAINING_MODE>OFF</TRAINING_MODE>
</RESPONSE>
```

WebSocket Connection

SCA is enhanced to use WebSocket to act as a client to an arbitrary broker. This is used to facilitate devices to maintain connections with a central web-based broker to enable POS clients communicating with those devices without having direct wired connections between device and POS. The communication of SCA with PWC and other hosts (and VHQ) should still happen over the normal TCP/IP connection.
WebSocket is applied only to the SCA API, normally used by the POS. POSCONNECTALLOW parameter is updated with a new value WEBSOCKETS to connect to POS via WebSocket. Since WebSocket will not use the MAC/MAC_LABEL, SKIPPOSAUTH parameter is added and if this is set, then it will bypass the SCI request validation. SCA has introduced other new parameters to configure WebSocket. Refer to WebSocket parameter table in SCA Engage Configuration Guide for more details.

WebSocket connection is supported in the following devices: e280, e285, M400, M440, P200, P400, V200, V400c and V400m.

- SCA Application continues trying the WebSocket connection to the POS cloud as part of the boot up process till it gets successful. SCA should display proper messages on screen while trying to connect and gives visual indication of retries and status of the connection on the screen. In case of any error, the messages will be displayed on screen and the status bar icons will be displayed to show current status of connection.

- SCA will support secondary over primary channel itself so that PSDK does not need to maintain two channels. SCA will listen for both primary and secondary port on single channel.
- SCA will NOT make REGISTRATION command as mandatory when mutual TLS authentication happens via WebSocket so that PSDK does not need to maintain the key for MACing purpose. If SKIPPOSAUTH parameter is set to 1 and POSCONNECTALLOW is set to WEBSCOKETS, then SCA will ignore MAC/MAC_LABEL etc. from POS Request and any SECURITY command would not be allowed.
- GET COUNTER command will be allowed. SCA will send back the COUNTER value sent in any last POS Request. If there is no last POS Request or GET COUNTER is the first command, then SCA will send back COUNTER as 1.
- In case of disconnection, SCA will re-establish the connection with POS Cloud server. Connection should be automatically recovered/re-connected if lost. Status bar icons will be changed accordingly. SCA will show the re-connecting status on screen as well if there is no open session. In case of any open session, only status bar icons will be updated.
- The below status bar icon location is /home/usr1/flash/www/images/wbs_disconnected.png for disconnected in terminal.
- The below status bar icon location is /home/usr1/flash/www/images/wbs_connected.png for connected in terminal. These can be updated/changed by providing same name icon files in /home/usr1/flash/www/images/ location.