



Lawful Interception

Version 21.6.1, 25 June 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

©Copyright 2022 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. Introduction	3
1.1. Supported Hardware Platforms	3
1.2. Components of Lawful Interception	3
2. LI Encapsulation	6
2.1. Packet Format Encoding	6
2.1.1. Payload Direction	6
2.1.2. Mapping Payload Format	6
2.1.2.1. Sub-payload Format (Type)	7
3. Enabling Lawful Interception	8
3.1. LI Request Object	10
3.2. Request Examples	10
3.2.1. Enabling LI via the BDS API	10
3.2.1.1. Adding LI	11
3.2.1.2. Deleting LI	11
3.3. Verification	11
3.4. RADIUS Lawful Interception	11

1. Introduction

Lawful Interception (LI) is a legal requirement in most of the countries. It enables the legal authorities to obtain communications network data for analysis or evidence. It is a method of intercepting certain data-streams of end-users in both directions, and tunnel the intercepted traffic to a Mediation Device (MD) with information about direction of capture and reference to the intercepted connection.

Leaf node is the Point of Interception (POI) and MD is the final Point of Collection (POC).

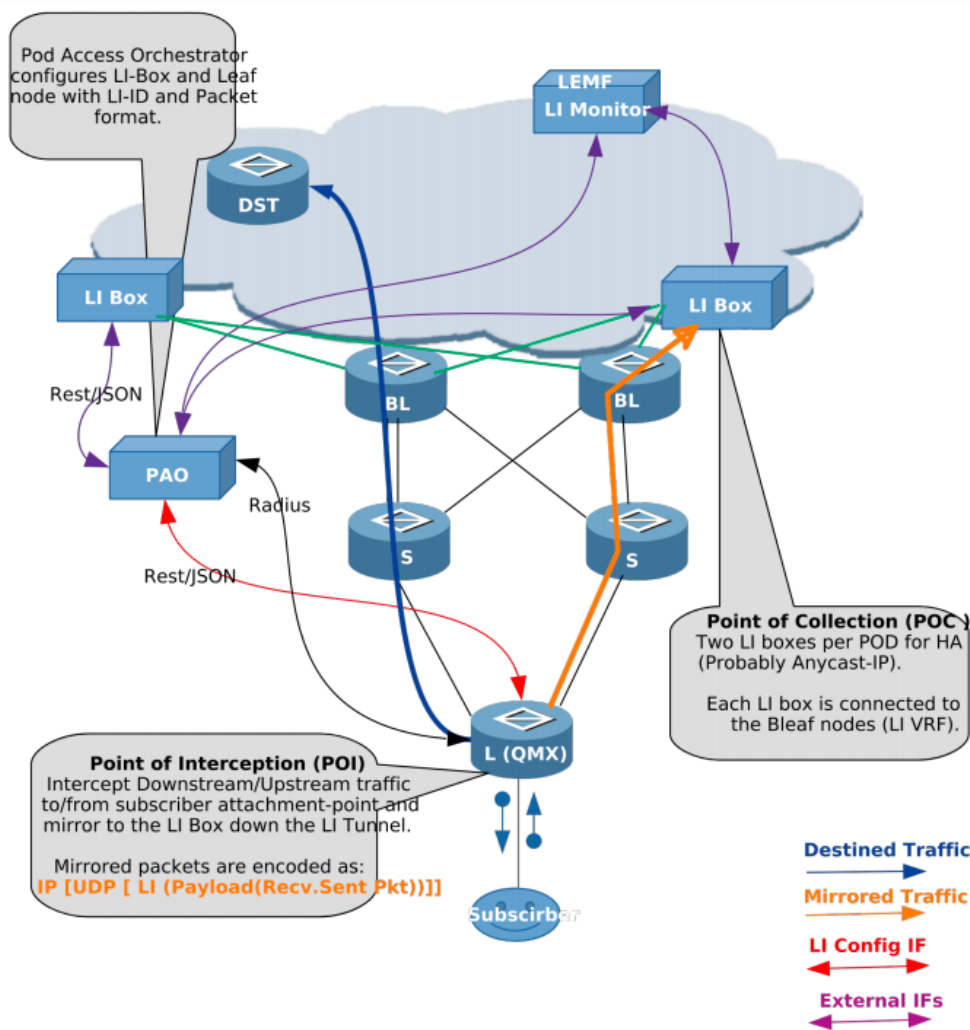
1.1. Supported Hardware Platforms

LI is supported on the following hardware platform:

- Broadcom's Qumran Switch

1.2. Components of Lawful Interception

The figure below shows the different components of the LI solution.



Definitions

L(QMX)

Leaf node in the POD which is connected to subscribers.

S/BL

Spine and Border Leaf in the POD, which can be replaced with just one node.

LI Box

Lawful Intercept Box, which communicates to Law Enforcement Agency (LEMF) and relays mirrored traffic. Two LI boxes per POD are connected for redundancy.

PAO

POD Access Orchestrator, which configures the LI Box and network nodes with LI configurations.

DST

Destination node for traffic from subscribers.

Abbreviations

Abbreviation	Definition
LI	Lawful Interception
POI	Point of Interception
POC	Point of Collection
PAO	Pod Access Orchestrator
LIMS	Lawful Interception Management System
VRF	Virtual Routing Instance
LEMF	Lawful Enforcement Monitoring Facility
Leaf	Access node
PPPoE	Point to Point Protocol over Ethernet
L2TP	Layer 2 Tunnelling Protocol
MPLS	Multi Protocol Label Switching

2. LI Encapsulation

Qumran-MX (BCM) supports LI with 32 bits shim header: SHIMoUDPoIPoETH

2.1. Packet Format Encoding

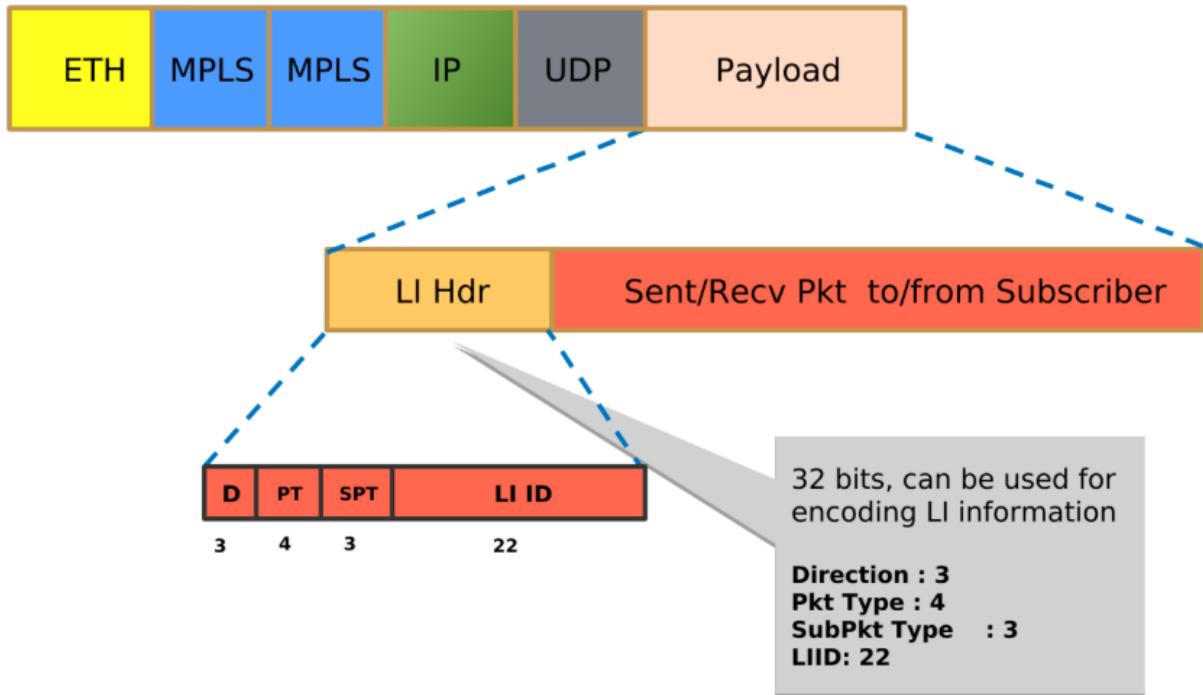


Figure 1. Packet Format Encoding

2.1.1. Payload Direction

Value	Payload Direction
0	Reserved for keepalive mechanism
2	Intercepted data or event was sent to target (downstream)
3	Intercepted data or event was sent from target (upstream)

2.1.2. Mapping Payload Format

Value	Payload Format
0-3	Reserved (unused)
4	Unknown, Not able to decide the PT
5	IPv4 packet (not used)
6	IPv6 packet (not used)
7	Ethernet Frame (used for Lawful Interception)

2.1.2.1. Sub-payload Format (Type)

The sub-payload formats are:

1. Single VLAN tag
2. Double VLAN tag
3. Untagged

3. Enabling Lawful Interception



- RBFS hides information about lawful intercepts from all but the most privileged users. An administrator must set up access rights to enable privileged users to access lawful intercept information.
- LI can be enabled for both L2TP and PPPoE subscribers.

The following example shows the **li_request_object** that allows you to create an add or a delete request.

```

{
  "object": {
    "codepoint" : 1302,
    "name"      : "li_request_object",
    "module"    : "subscriberd",
    "description": "An Object for LI add/delete request"
    "attribute": [

      {
        "mandatory" : false,
        "codepoint" : 1,
        "type"       : "uint64",
        "name"       : "subscriber_id",
        "description": "Subscriber Identifier"
      },
      {
        "mandatory" : true,
        "codepoint" : 2,
        "type"       : "uint32",
        "name"       : "li_id",
        "description": "LI Identifier (1-4194303(0x1FFFF))"
      },
      {
        "mandatory" : false,
        "codepoint" : 3,
        "type"       : "li_direction_map",
        "name"       : "li_direction",
        "description": "LI direction up(ingress)/down(egress)/both"
      },
      {
        "mandatory" : true,
        "codepoint" : 4,
        "type"       : "string",
        "name"       : "med_instance",
        "description": "VRF instance to lookup LI destination"
      },
      {
        "mandatory" : true,
        "codepoint" : 5,
        "type"       : "ipv4addr",
        "name"       : "med_ip",
        "description": "IP address of MD(LIBox)"
      },
      {
        "mandatory" : true,
        "codepoint" : 6,
        "type"       : "uint16",
        "name"       : "med_port",
        "description": "UDP port(MD)(49152-65535), mirrored traffic is forwarded"
      }
    ]
  }
}

```

The table below lists the parameters of **li_request_object**.

Parameter Name	Description
subscriber_id	Subscriber identifier that is generated by RBFS, for example, 72339069014638701.

Parameter Name	Description
li_id	Identifier for Lawful Interception. This is unique Identifier used by mediation device to identify the intercepted subscriber. The range can be between 1 to 4194303.
li_direction	LI direction. Values are: INGRESS, EGRESS, BOTH.
med_instance	VRF instance through the which the mediation device is reachable.
med_ip	IPv4 address of the mediation device
med_port	UDP port(MD)(49152-65535), mirrored traffic is forwarded



All the attributes of *li_request_object* are mandatory to configure LI.

3.1. LI Request Object

The *li_request_object* resides in the *secure.lawful.access.1.li_request* table.

The LI Request Object configuration is just a wrapper for the actual LI request table (**secure.lawful.access.1.li_request**) which is filled via RADIUS attributes. The required RADIUS attributes are already defined in the RADIUS dictionary. The **secure.lawful.access.1.li_request** table is protected by the secure management plane.

3.2. Request Examples

3.2.1. Enabling LI via the BDS API

The following example shows how to enable LI via the BDS API.

```
{
  "table": {
    "table_name": "secure.lawful.access.1.li_request"
  },
  "objects": [
    {
      "attribute": {
        "subscriber_id": "72339069014639042",
        "li_id": "66666",
        "li_direction": "BOTH",
        "med_instance": "libox",
        "med_ip": "10.0.0.1",
        "med_port": "49153"
      }
    }
  ]
}
```

3.2.1.1. Adding LI

The example below shows how to add the LI request via curl.

```
curl -i -H "Content-Type: application/json" -X POST -d @li_request.json
http://192.168.1.120:19091/api/v1/rbfs/elements/rtbrick/services/subscriberd.
1/proxy/bds/object
```

3.2.1.2. Deleting LI

The example below shows how to delete the LI request via curl.

```
curl -i -H "Content-Type: application/json" -X DELETE -d @li_request.json
http://192.168.1.120:19091/api/v1/rbfs/elements/rtbrick/services/subscriberd.
1/proxy/bds/object
```

3.3. Verification

The following commands allow you to view the table information. These are platform-specific tables.

```
show datastore table dump local.bcm.qmx.mirror
```

```
show datastore table dump local.bcm.qmx.acl
```

3.4. RADIUS Lawful Interception

All of the following attributes must be present in RADIUS access-accept or CoA request to control Lawful Interception (LI) via RADIUS. Those attributes are salt encrypted using the algorithm described in RFC 2868 for the Tunnel-Password. This encryption algorithm is defined for RADIUS access-accept messages only. To support CoA requests the request authenticator should be replaced with 16 zero bytes which is common industry standard.

The LI action NOOP can be used to obfuscate lawful interception requests (fake requests) to prevent that just the presence of those attributes indicates that a subscriber is intercepted. LI requests via RADIUS will show up in the same table as requests via REST or HTTP RPC API (secure.lawful.access.1.li_request).



The failed LI activations are not signalled via RADIUS to prevent that just the presence of CoA response NAK shows that LI request is not fake (action NOOP).

VSA 26-50058-140 - RtBrick-LI-Action (salt encrypted integer)

Value	Code	Description
NOOP	0	No action / Ignore LI request
ON	1	Start LI / Add LI request
OFF	2	Stop LI / Delete LI request

VSA 26-50058-141 - RtBrick-LI-Identifier (salt encrypted integer)

Device unique lawful interception identifier (LIID) within the range from 1 to 4194303.

VSA 26-50058-142 - RtBrick-LI-Direction (salt encrypted integer)

Value	Code	Description
INGRESS	1	Ingress mirroring only (from subscriber)
EGRESS	2	Egress mirroring only (to subscriber)
BOTH	3	Bidirectional mirroring (from and to subscriber)

VSA 26-50058-143 - RtBrick-LI-MED-Instance (salt encrypted string)

Routing instance through which the mediation device is reachable.

VSA 26-50058-144 - RtBrick-LI-MED-IP (salt encrypted IPv4 address)

IPv4 address of the mediation device.

VSA 26-50058-145 - RtBrick-LI-MED-Port (salt encrypted integer)

UDP port between 49152 and 65535 set in the mirrored traffic