Lawful Interception

Version 20.8.1, 10 August 2020

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

Table of Contents

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).
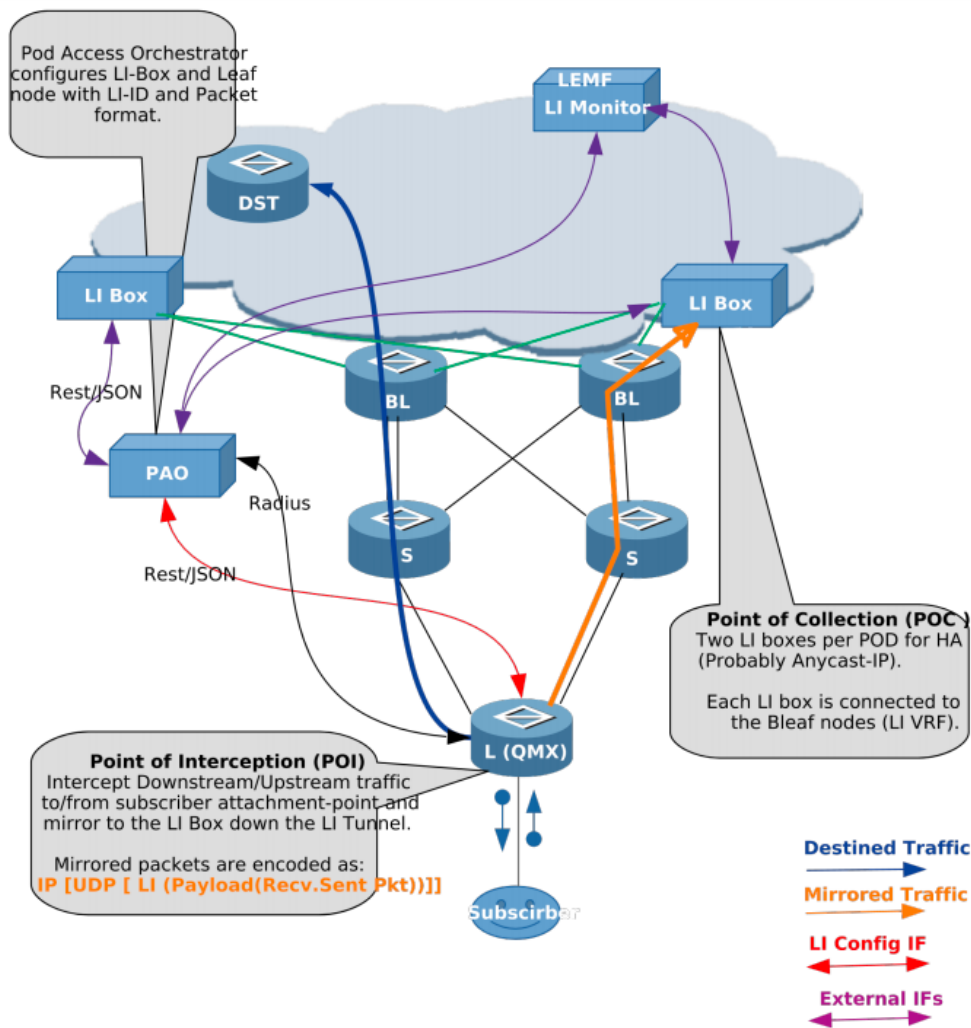
1.1. Supported Hardware Platforms

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
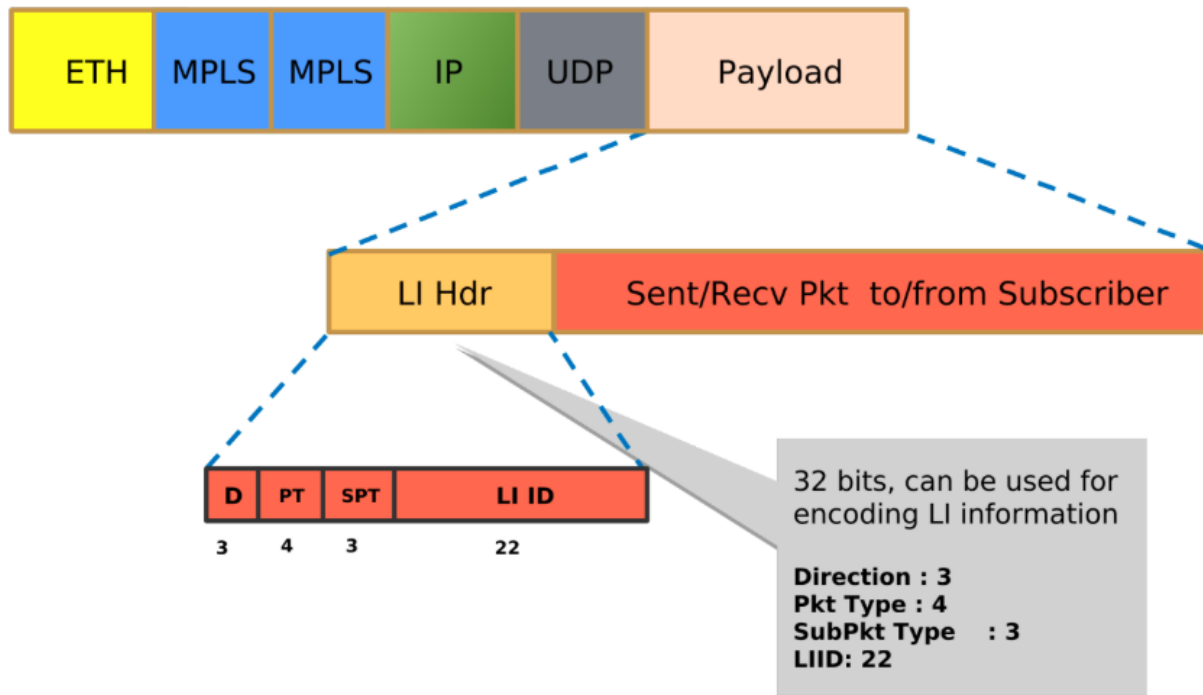
Figure 1. Packet Format Encoding

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 (used for upstream flow)
5	IPv4 packet (not used)
6	IPv6 packet (not used)
7	Ethernet Frame (used for downstream flow)

2.1.2.1. Sub-payload Format (Type)

The sub-payload formats are:

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

3. Configuring 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.

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. 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

3.1. LI Request Object

The *li_request_object* resides in the *global.access.1.li_request* table.

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

3.2. Configuration Example

The following example shows REST/JSON way of configuration LI.

```
{
    "table": {
        "table_name": "global.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"
            }
        }
    ]
}
```

3.3. 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