SD45N

Troubleshooting Guide – SD45N Linux Platform

Application Note *v. 1.1*

INTRODUCTION

The Laird SD45N (MSD45N and SSD45N) Linux release package must be integrated into your Linux host platform. This application note explains how to gather the necessary information so that you can troubleshoot the Wi-Fi solution by interpreting log information and, if necessary, forward the necessary information to Laird support.

Three main parts of events/logs of your Linux platform are most helpful for troubleshooting with Laird's release package.

- Laird Events
- dmesg log
- supplicant log

LAIRD EVENTS

The networking stack components send out event messages when certain things happen. This event system is designed to enable a network management application to be written to manage the state of the radio. Laird's events program example (event_mon) which runs on your host platform, is required. It routes the event messages to the syslog or the console. When your system boots, it must start the event_mon program in the background and log events to the syslog.

To monitor all events from Laird's SDK, we strongly recommend that you have event_mon integrated into your Linux host platform. Alternatively, you can also implement Laird Events into your own custom application. For implementing a custom solution, refer to Laird's SDK documentation.

The event_mon source package can be retrieved from:

https://github.com/LairdCP/wb-event_mon

Laird recommends running this application with the following options **-o logging -b 0x0000003FA3008000** and running it in the background.

```
event_mon -ologging -b0x0000003FA3008000 &
```

-b 0x0000003FA3008000 is a bitmask indicating the monitoring of the following events:

SDC_E_CONNECTION_STATE
SDC_E_DHCP
SDC_E_READY
SDC_E_ROAM
SDC_E_SCAN_REQ
SDC_E_DISCONNECT_REQ
SDC_E_SCAN
SDC_E_REGDOMAIN
SDC_E_CMDERROR
SDC_E_INTERNAL
SDC_E_FW_ERROR

The **-o** option allows you to control where logging is sent. For example:

- To log to console: **-o console**
- To log to syslog: **-o logging**
- To log both: **-o both**

To view additional event_mon options, use **even_mon -h**, **event_mon --help**, or view the source for the usage() function.

To enable DHCP events, refer to the **Laird Wi-Fi SDK Programmer's Guide**. The *Events (Linux only feature)* and *Implementing DHCP Even*ts sections provide additional integration information.

The following is an example of event_mon messages for a successful connection using LEAP authentication via a logread dump of syslog:

```
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_CONNECT_REQ Auth type:
AUTH_OPEN
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
ASSOCIATING
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
ASSOCIATED
Jan 1 00:01:27 summit user.info laird[625]: AP Mac address: 54:78:1a:42:b0:d7
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
AUTHENTICATING
Jan 1 00:01:27 summit user.info laird[625]: Auth reason: AUTH_REASON_UNSPEC
Jan 1 00:01:27 summit user.info laird[625]: AP Mac address: 54:78:1a:42:b0:d7
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
AUTHENTICATED
Jan 1 00:01:27 summit user.info laird[625]: Auth reason: AUTH_REASON_UNSPEC
Jan 1 00:01:27 summit user.info laird[625]: AP Mac address: 54:78:1a:42:b0:d7
Jan 1 00:01:27 summit user.info laird[625]: Event: SDC_E_DHCP status: REQUESTING
reason: DHCP_REASON_UNSPEC
Jan 1 00:01:27 summit user.info laird[625]: AP Mac address: 54:78:1a:42:b0:d7
Jan 1 00:01:28 summit user.info laird[625]: Event: SDC_E_DHCP status: BOUND reason:
IP_ADDRESS_DIFFERENT
Jan 1 00:01:28 summit user.info laird[625]: interface: wlan0
Jan 1 00:01:28 summit user.info laird[625]: address: 10.1.44.114
Jan 1 00:01:28 summit user.info laird[625]: subnet_mask: 255.255.255.0
Jan 1 00:01:28 summit user.info laird[625]: routers: 10.1.44.1
Jan 1 00:01:28 summit user.info laird[625]: lease_time: 14400
Jan 1 00:01:28 summit user.info laird[625]: message_type: 5
Jan 1 00:01:28 summit user.info laird[625]: dns_servers: 10.1.44.2 10.1.44.32
Jan 1 00:01:28 summit user.info laird[625]: dhcp_server: 1.1.1.1
Jan 1 00:01:28 summit user.info laird[625]: domain_name: akron.com
Jan 1 00:01:28 summit user.info laird[625]: renew: 4 1970/01/01 02:01:28
Jan 1 00:01:28 summit user.info laird[625]: rebind: 4 1970/01/01 03:31:28
Jan 1 00:01:28 summit user.info laird[625]: expire: 4 1970/01/01 04:01:28
Jan 1 00:01:28 summit user.info laird[625]: AP Mac address: 54:78:1a:42:b0:d7
```

Below is an example of a failed connection attempt with LEAP using the wrong password:

```
Jan 1 00:03:56 summit user.info laird[625]: Event: SDC_E_CONNECT_REQ Auth type:
AUTH_OPEN
Jan 1 00:03:56 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
ASSOCIATING
Jan 1 00:03:56 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
ASSOCIATED
Jan 1 00:03:56 summit user.info laird[625]: AP Mac address: 00:26:cb:f3:8c:09
Jan 1 00:03:56 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
AUTHENTICATING
Jan 1 00:03:56 summit user.info laird[625]: Auth reason: AUTH_REASON_UNSPEC
Jan 1 00:03:56 summit user.info laird[625]: AP Mac address: 00:26:cb:f3:8c:09
Jan 1 00:03:57 summit user.info laird[625]: Event: SDC_E_CONNECTION_STATE status:
AUTH_ERROR
Jan 1 00:03:57 summit user.info laird[625]: Auth reason: INVALID_CREDENTIALS
Jan 1 00:03:57 summit user.info laird[625]:  AP Mac address: 00:26:cb:f3:8c:09
```

A typical event_mon logging message contains and Event, Status, Reason information. Once the client associates to an AP, the AP mac address will then be printed. For SDC_E_DHCP events that indicate a status of BOUND, RENEWED, DECONFIG, or RELEASED, an output of the DHCP leases file will also be shown.

The following are the descriptions of events that are implemented.

Event	Description	
SDC_E_CONNECTION_STATE	A change to the Wi-Fi's connection state has occurred. See:	
	LRD_WF_EvtConStatus	status field
	LRD_WF_EvtAuthReason or SDC_ATH_DISCONNECT_REASON	reason field
	802.11 reason codes	auth_type field
SDC_E_DHCP	Indicates a DHCP event has occurred. Note that on systems using the SD45 see section on dhcp_injector and reason code to implement this event. If no implementation of dhcp_injector, this part of event can be ignored. See:	
	LRD_WF_EvtDHCPStatus	status field
	LRD_WF_EvtDHCPReason	reason field
SDC_E_READY	Indicates the wireless device is ready. This is sent once after a power-on or reset and after firmware recovery.	
SDC_E_CONNECT_REQ	Indicates a request to connect to a network.	
SDC_E_RECONNECT_REQ	Indicates a request to reconnect to a network to which the device was previously connected.	
SDC_E_DISCONNECT_REQ	Indicates a request to disconnect from a network.	
SDC_E_ASSOC	Indicates that a connection to a network has occurred.	
SDC_E_AUTH	Indicates that the authentication state has changed. See:	
	LRD_WF_EvtAuthStatus	status field
	LRD_WF_EvtAuthReason	reason field
SDC_E_DISASSOC	Indicates that the device has lost connectivity to a network or failed to associate. See:	
	SDC_ATH_DISCONNECT_REASON	status field
	802.11 reason codes	reason field
SDC_E_ROAM	Indicates a roam has occurred.	
SDC_E_SCAN_REQ	Indicates a request to initiate a scan from the host.	
SDC_E_SCAN	Indicates a host-initiated scan is complete. Check the status field for scan success or failure.	
SDC_E_REGDOMAIN	Indicates the firmware's regulatory domain has changed.	
SDC_E_CMDERROR	Indicates the firmware has reported an error. See:	
	SDC_ATH_CMDERROR_REASON	status field
SDC_E_INTERNAL	Indicates a status update or error from within the SDK events. See:	
	LRD_WF_EvtIntStatus	status field
	LRD_WF_EvtIntReason	reason field

Event	Description	
SDC_E_FW_ERROR	Indicates a firmware crash has occurred. If recovery is enabled, the event to indicate the firmware was recovered is SDC_E_READY. See:	
	LRD_WF_EvtFwErrorReason	reason field

LRD_WF_EvtConStatus

Reason	Description	
CON_STATUS_UNSPEC	The status is unknown.	
NOT_CONNECTED	The device is not currently connected.	
ASSOCIATING	The device is associating to the network.	
ASSOCIATED	The device is associated to the network.	
ASSOC_ERROR	There was an error while associating. See:	
	SDC_ATH_DISCONNECT_REASON	reason field
	802.11 reason codes	auth_type field
AUTHENTICATING	The device is authenticating.	
AUTHENTICATED	The device is authenticated.	
AUTH_ERROR	There was an error while authenticating. See:	
	LRD_WF_EvtAuthReason	reason field

SDC_ATH_DISCONNECT_REASON

Reason	Description
DISCON_REASON_UNSPEC	No reason specified.
NO_NETWORK_AVAIL	Unable to find or establish a connection to the desired network.
LOST_LINK	Missed too many beacons.
DISCONNECT_CMD	A disconnect request was processed.
BSS_DISCONNECTED	The device is on an AP blacklist (MAC block) or not on the AP whitelist, the AP is too busy to accept connections or too many encryption errors have occurred.
AUTH_FAILED	Not used.
ASSOC_FAILED	Not used.
NO_RESOURCES_AVAIL	The firmware is out of memory.
CSERV_DISCONNECT	The firmware has decided to disconnect from network. This can occur from host-influenced settings such as marking an AP as *bad* or because there have been too many decryption errors. If in Ad-Hoc mode, the firmware does not see the other client.
INVALID_PROFILE	The host sent a bad BSSID.
DOT11H_CHANNEL_SWITCH	The AP sent a DOT11H CSA IE (802.11h Channel Switch Announcement).
PROFILE_MISMATCH	Occurs if the device is in ad-hoc mode and powersave is enabled.
CONNECTION_EVICTED	Not used.
IBSS_MERGE	The station has merged with another IBSS.

802.11 Reason Codes

Reason	Code	Description
DOT11_RC_RESERVED	0	Reserved
DOT11_RC_UNSPECIFIED	1	Indicates an unspecified reason.
DOT11_RC_AUTH_INVAL	2	Indicates that the previous authentication is no longer valid.
DOT11_RC_DEAUTH_LEAVING	3	Indicates a deauthentication because the sending station is leaving (or has left) IBSS or ESS.
DOT11_RC_INACTIVITY	4	Indicates a disassociation due to inactivity.
DOT11_RC_BUSY	5	Indicates a disassociation because the AP is unable to handle all currently associated stations.
DOT11_RC_INVAL_CLASS_2	6	Indicates that a Class 2 frame was received from a non-authenticated station.
DOT11_RC_INVAL_CLASS_3	7	Indicates that a Class 3 frame was received from a non-authenticated station.
DOT11_RC_DISASSOC_LEAVING	8	Indicates a disassociation because the sending station is leaving (or has left) BSS.
DOT11_RC_NOT_AUTH	9	Indicates that the station that is requesting (re)association is not authenticated with the responding station.
DOT11_RC_BAD_PC	10	Indicates an unacceptable power capability element.
DOT11_RC_BAD_CHANNELS	11	Indicates an unacceptable supported channels element.
DOT11_RC_BSS_TRANSIT_MGMT	12	Indicates a disassociation due to BSS Transition Management.
DOT11_RC_INVALID_WPA_IE	13	Indicates an invalid info. element.
DOT11_RC_MIC_FAILURE	14	Indicates a Michael failure.
DOT11_RC_4WH_TIMEOUT	15	Indicates a four-way handshake timeout.
DOT11_RC_GTK_UPDATE_TIMEOUT	16	Indicates a group key update timeout.
DOT11_RC_WPA_IE_MISMATCH	17	Indicates that a WPA IE in a four-way handshake differs from a (re)association request/probe response.
DOT11_RC_INVALID_MC_CIPHER	18	Indicates an invalid multicast cipher.
DOT11_RC_INVALID_UC_CIPHER	19	Indicates an invalid unicast cipher.
DOT11_RC_INVALID_AKMP	20	Indicates an invalid authenticated key management protocol.
DOT11_RC_BAD_WPA_VERSION	21	Indicates an unsupported WPA version.
DOT11_RC_INVALID_WPA_CAP	22	Indicates invalid WPA IE capabilities.
DOT11_RC_8021X_AUTH_FAIL	23	Indicates an 802.1X authentication failure.
DOT11_RC_UNSPECIFIED_QOS	32	Indicates an unspecified QoS-related reason.
DOT11_RC_INSUFFICIENT_BW	33	Indicates that the QoS AP lacks sufficient bandwidth for this QoS station.
DOT11_RC_EXCESSIVE_FRAMES	34	Indicates that excessive frames need to be acknowledged due to AP transmissions or poor channel conditions.
DOT11_RC_TX_OUTSIDE_TXOP	35	Indicates that the station is transmitting outside the limits of its TXOPs.
DOT11_RC_LEAVING_QBSS	36	Indicates a request from the peer station as the station is leaving the BSS (or resetting).

Reason	Code	Description
DOT11_RC_BAD_MECHANISM	37	Indicates a request from the peer station that it does not want to use the mechanism.
DOT11_RC_SETUP_NEEDED	38	Indicates a request from the peer station that the station received frames using the mechanism that require setup.
DOT11_RC_TIMEOUT	39	Indicates a request from the peer station that there was a timeout.

LRD_WF_EvtAuthStatus

Reason	Description
AUTH_STATUS_UNSPEC	Status not specified.
AUTH_STARTED	Authentication started.
AUTH_SUCCESS	Authentication succeeded.
AUTH_FAILURE	Authentication failed. See: LRD_WF_EvtAuthReason reason field

LRD_WF_EvtAuthReason

Reason	Description
AUTH_REASON_UNSPEC	The reason is unspecified.
AUTH_SERVER_NO_RESP	Indicates that there was no response from the RADIUS server. This can indicate the RADIUS server did not respond, the connection is very poor, or the connection was too short to receive a response.
INVALID_CREDENTIALS	Indicates that the credentials are invalid.
METHOD_NOT_SUPPORTED	Indicates that the authentication method is not supported by the RADIUS server.
INVALID_CERT_PASS	Indicates that the certificate password is invalid.

SDC_ATH_CMDERROR_REASON:

Reason	Description
INVALID_PARAM	An invalid parameter was sent to the firmware.
ILLEGAL_STATE	The firmware is in an illegal state.
INTERNAL_ERROR	An internal error has occurred in the firmware.

LRD_WF_EvtIntStatus:

Reason	Description
INT_STATUS_UNSPEC	Status is not specified.
LOST_COM_DRV	Lost communication with the driver.
LOST_COM_KERN	Lost communication with the kernel.
LOST_COM_SUPP	Lost communication with the supplicant.
LOST_COM_INJ	Lost communication with injected events.

LRD_WF_EvtIntReason

Reason	Description
INT_REASON_UNSPEC	Reason is not specified.

COM_EXITED	Lost communication due to the other side exiting.
COM_ERROR	Lost communication due to error.

LRD_WF_EvtFwErrorReason

Reason	Description
FW_ASSERT	Firmware asserted.
FW_HB_RESP_FAILURE	Firmware did not respond to enough heartbeats.
FW_EP_FULL	Firmware stopped servicing firmware commands.

LRD_WF_EvtDHCPStatus

Reason	Description
DHCP_STATUS_UNSPEC	Indicates that the status is not specified.
DECONFIG	The DHCP has requested that the interface configuration be removed.
REQUESTING	Indicates that the Disover was sent and the DHCPOFFER replay was received.
RENEWING	Indicates that half of the lease was passed or that the station has reconnected to the network and wants to renew. A unicast renew request is being sent.
RENEWED	Indicates that the lease has renewed. See: LRD_WF_EvtDHCPReason | reason field
REBINDING	Indicates that the renew requests were not answered and a broadcast renew is being sent.
BOUND	Indicates that a select/renew was sent and a DHCPACK reply was received. The interface will be configured with lease. See: LRD_WF_EvtDHCPReason | reason field
NAK	Indicates that Nak was received from the server.
LEASEFAIL	Indicates that the DHCP client has failed to obtain a lease.
RELEASED	Indicates that the DHCP client has sent a release.

LRD_WF_EvtDHCPReason

Reason	Description
DHCP_REASON_UNSPEC	The reason is not specified.
IP_ADDRESS_SAME	Indicates that the IP address is the same as the previous lease.
IP_ADDRESS_DIFFERENT	Indicates that the IP address is different from the previous lease.

DMESG LOG

The kernel log is read by the program **dmesg**.

All subsystems in the kernel use the log, printing messages to it. The messages are labeled with a priority and usually with what subsystem it came from.

Messages are pre-pended with the subsystem it comes from.

Important subsystems:

- **ath6kl –** The Wi-Fi chip driver for the MSD45. If you are having trouble with the Wi-Fi chip specifically, these messages are keys.
- **cfg8021 –** This is the subsystem that is responsible for WiFi configuration parameters in the kernel.
- **mmc0 –** This is the MMC card and SDIO driver subsystem. Look here if you're having low-level SDIO problems.

The output should resemble the following:

```
usb0: MAC 7e:07:10:8c:e5:e5
usb0: HOST MAC 4a:24:fc:b7:0d:41
gadget: Ethernet Gadget, version: Memorial Day 2008
gadget: g_ether ready
ath6kl: INIT GENERIC NETLINK ATHEROS COM
ath6kl: ar6003 hw 2.1.1 sdio fw 3.4.0.0084\x01 api 4
nf_conntrack version 0.5.0 (947 buckets, 3788 max)
macb f802c000.ethernet eth0: link up (100/Full)
ip_tables: (C) 2000-2006 Netfilter Core Team
cfg80211: Calling CRDA for country: US
cfg80211: Regulatory domain changed to country: US
cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp)
cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (300 mBi, 2700 mBm)
cfg80211: (5170000 KHz - 5250000 KHz @ 40000 KHz), (300 mBi, 1700 mBm)
cfg80211: (5250000 KHz - 5330000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
cfg80211: (5490000 KHz - 5600000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
cfg80211: (5650000 KHz - 5710000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
cfg80211: (5735000 KHz - 5835000 KHz @ 40000 KHz), (300 mBi, 3000 mBm)

cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 4000 mBm)
```

The debugging flags are values set as a bitmask in **drivers/net/wireless/ath/ath6kl/debug.h**:

```
enum ATH6K_DEBUG_MASK {
ATH6KL_DBG_CREDIT = BIT(0),
/* hole */
ATH6KL_DBG_WLAN_TX = BIT(2), /* wlan tx */
ATH6KL_DBG_WLAN_RX = BIT(3), /* wlan rx */
ATH6KL_DBG_BMI = BIT(4), /* bmi tracing */
ATH6KL_DBG_HTC = BIT(5),
ATH6KL_DBG_HIF = BIT(6),
ATH6KL_DBG_IRQ = BIT(7), /* interrupt processing */
/* hole */
/* hole */
ATH6KL_DBG_WMI = BIT(10), /* wmi tracing */
ATH6KL_DBG_TRC = BIT(11), /* generic func tracing */
ATH6KL_DBG_SCATTER = BIT(12), /* hif scatter tracing */
ATH6KL_DBG_WLAN_CFG = BIT(13), /* cfg80211 i/f file tracing */
ATH6KL_DBG_RAW_BYTES = BIT(14), /* dump tx/rx frames */
ATH6KL_DBG_AGGR = BIT(15), /* aggregation */
ATH6KL_DBG_SDIO = BIT(16),
ATH6KL_DBG_SDIO_DUMP = BIT(17),
ATH6KL_DBG_BOOT = BIT(18), /* driver init and fw boot */
ATH6KL_DBG_WMI_DUMP = BIT(19),
ATH6KL_DBG_SUSPEND = BIT(20),
ATH6KL_DBG_USB = BIT(21),
ATH6KL_DBG_USB_BULK = BIT(22),
ATH6KL_DBG_RECOVERY = BIT(23),
ATH6KL_DBG_ANY = 0xffffffff /* enable all logs */
};
```

Combining the different bits enables varying debug outputs from the driver.

The debug_mask module parameter may be set either at the time you load the driver or during run time.

To set the debug mask on load:

```
# modprobe ath6kl_core debug_mask=0x00050000
# modprobe ath6kl_sdio
```

To set the debug mask at runtime:

```
# echo 0x00002400 > /sys/module/ath6kl_core/parameters/debug_mask
```

The kernel log messages output messages of WARNING and above only by default. At run time, you can change the minimum level that is output.

To determine your current console_loglevel, enter the following into the command console:

```
# cat /proc/sys/kernel/printk
```

The output should resemble the following:

```
4 7 1 7
```

The first integer of the output displays current console_loglevel; the second shows the default log level that you saw above.

To change your current console_loglevel, write to this file and change the settings. In order to get all messages printed to the console, enter the following into the command prompt:

```
# echo 8 > /proc/sys/kernel/printk
```

Once you've changed this, every kernel message appears on your console.

The log can be captured by redirecting the output of dmesg to a file as in the following:

```
#dmesg > /tmp/outputlog.txt
```

Use the parameter **–c** to clear the log after reading it if desired.

SUPPLICANT LOG

The supplicant does not have a log unless explicitly turning on debugging. Debugging is turned on with the **-d(d*)** switch where the more *d*s you add, the more debugs you get. For example, **-ddd** sets the debug level to level 3. Add this to the launch command of sdc_supp to get debug output. The output goes wherever you tell it. By default, debug is directed to **stdout**. To capture to a file, redirect the output file as desired. Alternatively, if you are using syslog, you can have the log sent there via the use of the **-s** option.

The following is an example that sets the debug to a typical level and sends it to syslog:

```
sdc_supp -iwlan0 -Dnl80211 -s -ddd &
```

Similarly, to initiate the log being sent to the /var/log/sdc_supp.txt file:

```
sdc_supp -iwlan0 -Dnl80211 -ddd > /var/log/sdc_supp.txt
```

Note: Because the log increases over time, if you use the second method above method, be aware that the file system holding **/var** is affected.

REVISION HISTORY

Revision	Date	Description	Approved By
1.0	Nov 2014	Initial Version	Brian Wagner
1.1	28 Apr 2015	Changed MSD45N to SD45N to incorporate the SSD45N	Sue White