

LANbeacon

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	open_ssl_keys Struct Reference . . . . .	5
3.2	received_lan_beacon_frame Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	5
3.2.2	Field Documentation . . . . .	6
3.2.2.1	challenge . . . . .	6
3.2.2.2	current_destination_mac . . . . .	6
3.2.2.3	lan_beacon_ReceivedPayload . . . . .	6
3.2.2.4	parsedBeaconContents . . . . .	6
3.2.2.5	payloadSize . . . . .	6
3.2.2.6	successfullyAuthenticated . . . . .	6
3.2.2.7	times_left_to_display . . . . .	7
3.3	receiver_information Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Field Documentation . . . . .	7
3.3.2.1	authenticated_mode . . . . .	8
3.3.2.2	current_lan_beacon_pdu_for_printing . . . . .	8
3.3.2.3	lanbeacon_keys . . . . .	8

3.3.2.4	<a href="#">my_receiver_interfaces</a>	8
3.3.2.5	<a href="#">number_of_currently_received_frames</a>	8
3.3.2.6	<a href="#">pointers_to_received_frames</a>	8
3.3.2.7	<a href="#">scroll_speed</a>	8
3.4	<a href="#">receiver_interfaces Struct Reference</a>	9
3.4.1	<a href="#">Detailed Description</a>	9
3.4.2	<a href="#">Field Documentation</a>	9
3.4.2.1	<a href="#">maxSockFd</a>	9
3.4.2.2	<a href="#">numInterfaces</a>	9
3.4.2.3	<a href="#">sockfd</a>	9
3.4.2.4	<a href="#">sockopt</a>	9
3.5	<a href="#">sender_information Struct Reference</a>	10
3.5.1	<a href="#">Detailed Description</a>	10
3.5.2	<a href="#">Field Documentation</a>	10
3.5.2.1	<a href="#">interface_to_send_on</a>	10
3.5.2.2	<a href="#">lan_beacon_pdu_len</a>	10
3.5.2.3	<a href="#">lanbeacon_keys</a>	11
3.5.2.4	<a href="#">lanBeacon_PDU</a>	11
3.5.2.5	<a href="#">send_frequency</a>	11
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>13</b>
4.1	<a href="#">/home/flopsstyle/gitLANbeacon/LANbeacon/src/define.h File Reference</a>	13
4.1.1	<a href="#">Detailed Description</a>	15
4.2	<a href="#">/home/flopsstyle/gitLANbeacon/LANbeacon/src/main.h File Reference</a>	15
4.2.1	<a href="#">Detailed Description</a>	16
4.2.2	<a href="#">Function Documentation</a>	16
4.2.2.1	<a href="#">main()</a>	16
4.3	<a href="#">/home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h File Reference</a>	16
4.3.1	<a href="#">Detailed Description</a>	18
4.3.2	<a href="#">Function Documentation</a>	18
4.3.2.1	<a href="#">bananaPIprint()</a>	18
4.3.2.2	<a href="#">evaluatelanbeacon()</a>	19
4.4	<a href="#">/home/flopsstyle/gitLANbeacon/LANbeacon/src/sender.h File Reference</a>	19
4.4.1	<a href="#">Detailed Description</a>	21
4.4.2	<a href="#">Function Documentation</a>	21
4.4.2.1	<a href="#">ipParser()</a>	21
4.4.2.2	<a href="#">mergedlanbeaconCreator()</a>	21
4.4.2.3	<a href="#">transferToCombinedBeacon()</a>	22
4.4.2.4	<a href="#">transferToCombinedBeaconAndString()</a>	22
4.4.2.5	<a href="#">transferToCombinedString()</a>	23
<b>Index</b>		<b>25</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">open_ssl_keys</a>	5
<a href="#">received_lan_beacon_frame</a>	
Contains all the information related to one received frame	5
<a href="#">receiver_information</a>	
Receiver configurations	7
<a href="#">receiver_interfaces</a>	
Contains all variables, that are needed to access sockets on interfaces	9
<a href="#">sender_information</a>	
Sender configurations	10



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">define.h</a>	
Contains application-wide includes with information such as addresses and TLV types . . . . .	13
/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">main.h</a>	
Main function and help function . . . . .	15
/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">openssl_sign.h</a> . . . . .	??
/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">rawsocket_LAN_Beacon.h</a> . . . . .	??
/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">receiver.h</a>	
Receiver-specific functions and structures . . . . .	16
/home/flopsstyle/gitLANbeacon/LANbeacon/src/ <a href="#">sender.h</a>	
Sender-specific functions and structures . . . . .	19





## Chapter 3

# Data Structure Documentation

### 3.1 open\_ssl\_keys Struct Reference

#### Data Fields

- char **path\_To\_Verifying\_Key** [KEY\_PATHLENGTH\_MAX+1]
- char **path\_To\_Signing\_Key** [KEY\_PATHLENGTH\_MAX+1]
- char **pcszPassphrase** [1024]
- int **generate\_keys**
- int **sender\_or\_receiver\_mode**

The documentation for this struct was generated from the following file:

- /home/flopsstyle/gitLANbeacon/LANbeacon/src/openssl\_sign.h

### 3.2 received\_lan\_beacon\_frame Struct Reference

Contains all the information related to one received frame.

```
#include <receiver.h>
```

#### Data Fields

- unsigned char **lan\_beacon\_ReceivedPayload** [LAN\_BEACON\_BUF\_SIZ]
- ssize\_t **payloadSize**
- unsigned long **challenge**
- unsigned char **current\_destination\_mac** [6]
- int **successfullyAuthenticated**
- int **times\_left\_to\_display**
- char \*\* **parsedBeaconContents**

#### 3.2.1 Detailed Description

Contains all the information related to one received frame.

### 3.2.2 Field Documentation

#### 3.2.2.1 challenge

```
unsigned long received_lan_beacon_frame::challenge
```

The challenge, that has been sent to the server.

#### 3.2.2.2 current\_destination\_mac

```
unsigned char received_lan_beacon_frame::current_destination_mac[6]
```

The MAC address of the server, which the frame was received from.

#### 3.2.2.3 lan\_beacon\_ReceivedPayload

```
unsigned char received_lan_beacon_frame::lan_beacon_ReceivedPayload[LAN_BEACON_BUF_SIZ]
```

Contains the raw received payload from a LAN-Beacon frame.

#### 3.2.2.4 parsedBeaconContents

```
char** received_lan_beacon_frame::parsedBeaconContents
```

Contains the parsed contents, that will be used to print something to the display.

#### 3.2.2.5 payloadSize

```
ssize_t received_lan_beacon_frame::payloadSize
```

The size of the raw payload.

#### 3.2.2.6 successfullyAuthenticated

```
int received_lan_beacon_frame::successfullyAuthenticated
```

Has frame already been authenticated?

## 3.2.2.7 times\_left\_to\_display

```
int received_lan_beacon_frame::times_left_to_display
```

Countdown, how many more times the frame will be displayed. Is updated, if frame with same content is received again.

The documentation for this struct was generated from the following file:

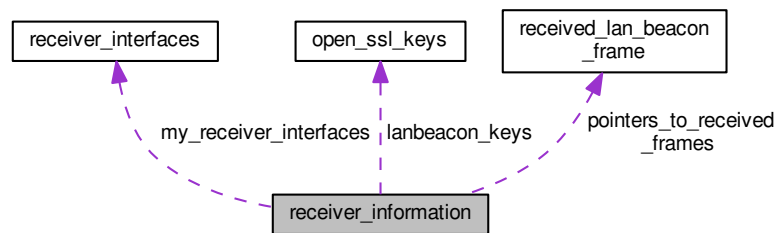
- </home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h>

## 3.3 receiver\_information Struct Reference

Receiver configurations.

```
#include <receiver.h>
```

Collaboration diagram for receiver\_information:



## Data Fields

- int [authenticated\\_mode](#)
- int [scroll\\_speed](#)
- int [current\\_lan\\_beacon\\_pdu\\_for\\_printing](#)
- struct [received\\_lan\\_beacon\\_frame](#) \* [pointers\\_to\\_received\\_frames](#) [20]
- int [number\\_of\\_currently\\_received\\_frames](#)
- struct [open\\_ssl\\_keys](#) [lanbeacon\\_keys](#)
- struct [receiver\\_interfaces](#) [my\\_receiver\\_interfaces](#)

## 3.3.1 Detailed Description

Receiver configurations.

## 3.3.2 Field Documentation

### 3.3.2.1 authenticated\_mode

```
int receiver_information::authenticated_mode
```

Has user specified using the authenticated mode?

### 3.3.2.2 current\_lan\_beacon\_pdu\_for\_printing

```
int receiver_information::current_lan_beacon_pdu_for_printing
```

The currently printed PDU.

### 3.3.2.3 lanbeacon\_keys

```
struct open_ssl_keys receiver_information::lanbeacon_keys
```

The paths to the keys.

### 3.3.2.4 my\_receiver\_interfaces

```
struct receiver_interfaces receiver_information::my_receiver_interfaces
```

Interfaces, that are used for LAN-Beacon reception.

### 3.3.2.5 number\_of\_currently\_received\_frames

```
int receiver_information::number_of_currently_received_frames
```

How many frames are currently stored for displaying.

### 3.3.2.6 pointers\_to\_received\_frames

```
struct received_lan_beacon_frame* receiver_information::pointers_to_received_frames[20]
```

Frames, that currently are stored for displaying.

### 3.3.2.7 scroll\_speed

```
int receiver_information::scroll_speed
```

How fast the display should switch to the next display page.

The documentation for this struct was generated from the following file:

- </home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h>

## 3.4 receiver\_interfaces Struct Reference

Contains all variables, that are needed to access sockets on interfaces.

```
#include <receiver.h>
```

### Data Fields

- int [sockfd](#) [20]
- int [sockopt](#) [20]
- int [maxSockFd](#)
- int [numInterfaces](#)

### 3.4.1 Detailed Description

Contains all variables, that are needed to access sockets on interfaces.

### 3.4.2 Field Documentation

#### 3.4.2.1 maxSockFd

```
int receiver_interfaces::maxSockFd
```

Needed for select function.

#### 3.4.2.2 numInterfaces

```
int receiver_interfaces::numInterfaces
```

Number of used interfaces.

#### 3.4.2.3 sockfd

```
int receiver_interfaces::sockfd[20]
```

File descriptors of raw sockets.

#### 3.4.2.4 sockopt

```
int receiver_interfaces::sockopt[20]
```

. Options for each raw socket.

The documentation for this struct was generated from the following file:

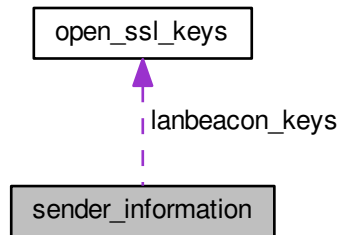
- [/home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h](#)

### 3.5 sender\_information Struct Reference

Sender configurations.

```
#include <sender.h>
```

Collaboration diagram for sender\_information:



#### Data Fields

- char \* [lanBeacon\\_PDU](#)
- int [lan\\_beacon\\_pdu\\_len](#)
- int [send\\_frequency](#)
- char \* [interface\\_to\\_send\\_on](#)
- struct [open\\_ssl\\_keys](#) [lanbeacon\\_keys](#)

#### 3.5.1 Detailed Description

Sender configurations.

#### 3.5.2 Field Documentation

##### 3.5.2.1 interface\_to\_send\_on

```
char* sender_information::interface_to_send_on
```

If specified, interface that is used for sending.

##### 3.5.2.2 lan\_beacon\_pdu\_len

```
int sender_information::lan_beacon_pdu_len
```

Length of the combined PDU.

### 3.5.2.3 lanbeacon\_keys

```
struct open_ssl_keys sender_information::lanbeacon_keys
```

Keys configuration.

### 3.5.2.4 lanBeacon\_PDU

```
char* sender_information::lanBeacon_PDU
```

The combined payload of a PDU, that is being sent.

### 3.5.2.5 send\_frequency

```
int sender_information::send_frequency
```

Number of seconds between each sent PDU.

The documentation for this struct was generated from the following file:

- [/home/flopsstyle/gitLANbeacon/LANbeacon/src/sender.h](#)





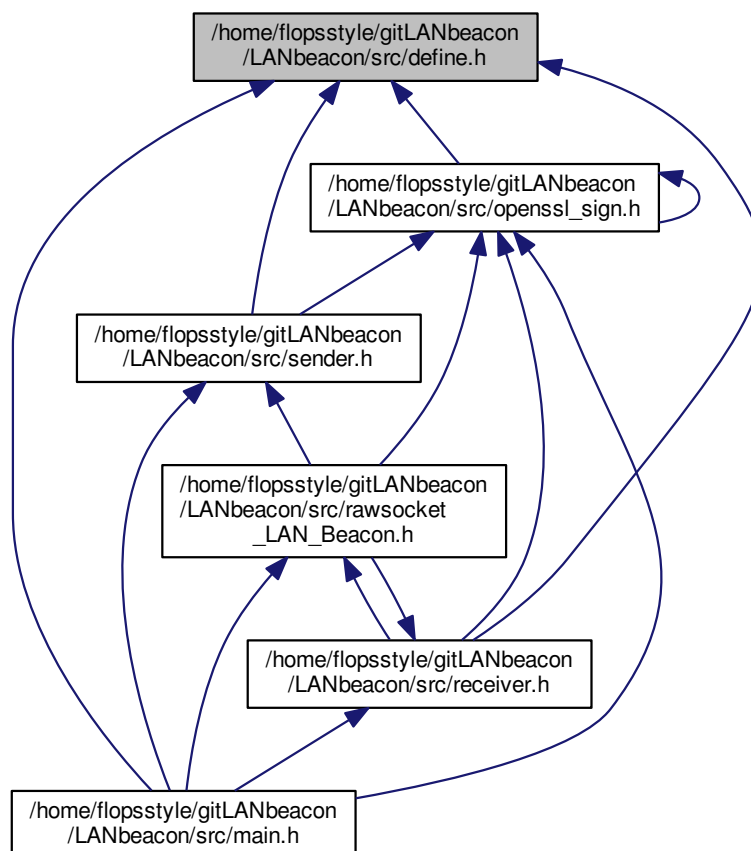
## Chapter 4

# File Documentation

### 4.1 /home/flopsstyle/gitLANbeacon/LANbeacon/src/define.h File Reference

Contains application-wide includes with information such as addresses and TLV types.

This graph shows which files directly or indirectly include this file:



## Macros

- `#define _(STRING) gettext(STRING) /** @name Macro for gettext localization support */`
- `#define LAN_BEACON_SEND_FREQUENCY 5 /** @name Send frequency in seconds */`

### LAN-Beacon Multicast addresses and EtherTypes

- `#define LAN_BEACON_DEST_MAC 0xff, 0xff, 0xff, 0xff, 0xff, 0xff`
- `#define LAN_BEACON_ETHER_TYPE 0x88B5`
- `#define CHALLENGE_ETHTYPE 0x88B6`

### Buffer sizes

- `#define PARSED_TLVs_MAX_NUMBER 25`
- `#define PARSED_TLVs_MAX_LENGTH 510`
- `#define LAN_BEACON_BUF_SIZ 2000`
- `#define KEY_PATHLENGTH_MAX 800`

### Standard paths

- `#define PRIVATE_KEY_STANDARD_PATH "private_key.pem"`
- `#define PUBLIC_KEY_STANDARD_PATH "public_key.pem"`

### Display options

- `#define DEFAULT_SCROLLSPEED 2`
- `#define SHOW_FRAMES_X_TIMES 3`
- `#define DESCRIPTOR_WIDTH 10`

### Subtype numbers lanbeacon

- `#define SUBTYPE_VLAN_ID 200`
- `#define SUBTYPE_NAME 201`
- `#define SUBTYPE_CUSTOM 202`
- `#define SUBTYPE_IPV4 203`
- `#define SUBTYPE_IPV6 204`
- `#define SUBTYPE_EMAIL 205`
- `#define SUBTYPE_DHCP 206`
- `#define SUBTYPE_ROUTER 207`
- `#define SUBTYPE_SIGNATURE 216`
- `#define SUBTYPE_COMBINED_STRING 217`

### Descriptor strings lanbeacon

- `#define DESCRIPTOR_VLAN_ID gettext("VLAN-ID:")`
- `#define DESCRIPTOR_NAME gettext("VLAN-Name:")`
- `#define DESCRIPTOR_CUSTOM gettext("Freetext:")`
- `#define DESCRIPTOR_IPV4 gettext("IPv4:")`
- `#define DESCRIPTOR_IPV6 gettext("IPv6:")`
- `#define DESCRIPTOR_EMAIL gettext("Email:")`
- `#define DESCRIPTOR_DHCP gettext("DHCP:")`
- `#define DESCRIPTOR_ROUTER gettext("Router:")`
- `#define DESCRIPTOR_SIGNATURE gettext("Authentication:")`
- `#define DESCRIPTOR_COMBINED_STRING gettext("Combined String:")`

### 4.1.1 Detailed Description

Contains application-wide includes with information such as addresses and TLV types.

#### Author

Dominik Bitzer

#### Date

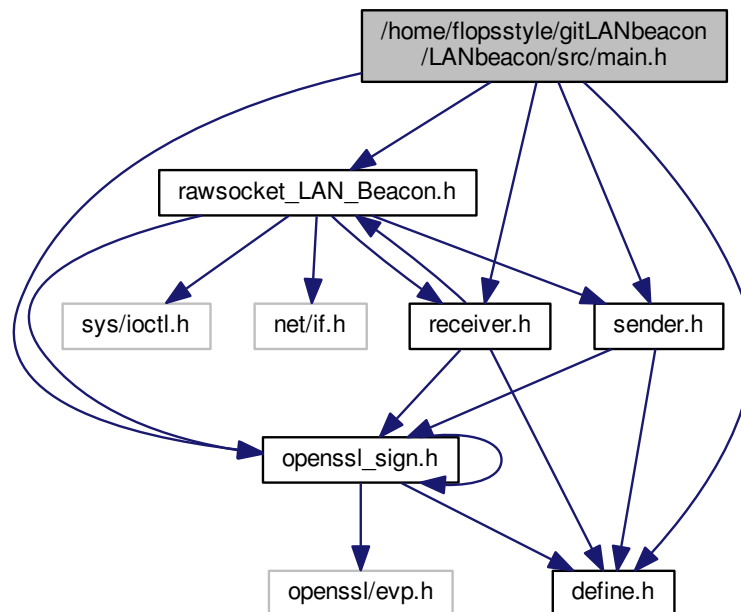
2017

## 4.2 /home/flopsstyle/gitLANbeacon/LANbeacon/src/main.h File Reference

Main function and help function.

```
#include "openssl_sign.h"
#include "sender.h"
#include "rawsocket_LAN_Beacon.h"
#include "receiver.h"
#include "define.h"
```

Include dependency graph for main.h:



### Functions

- int `main` (int argc, char \*\*argv)  
Separates receiver from sender mode and has the main program logic.
- void `printHelp` ()  
Help function, executed if unknown parameters have been received or user specifically asks for help.

### 4.2.1 Detailed Description

Main function and help function.

#### Author

Dominik Bitzer

#### Date

2017

### 4.2.2 Function Documentation

#### 4.2.2.1 main()

```
int main (
    int argc,
    char ** argv )
```

Separates receiver from sender mode and has the main program logic.

#### Returns

Success or failure code.

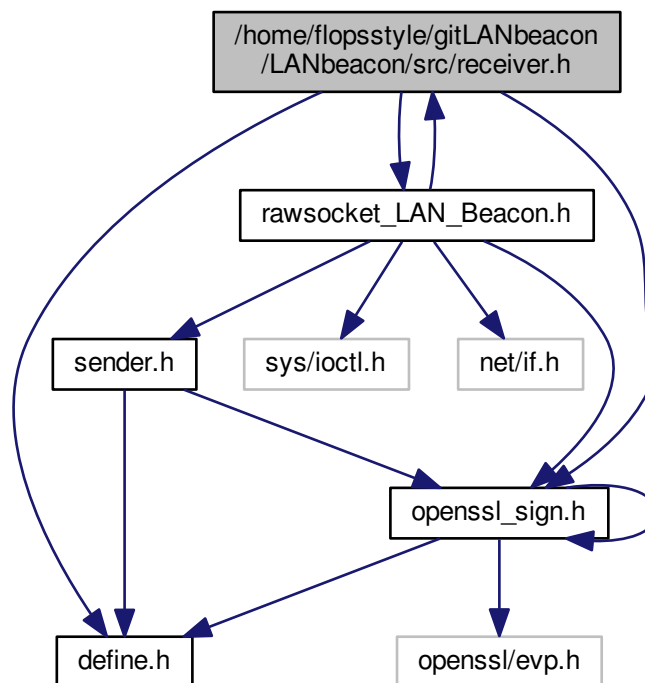
## 4.3 /home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h File Reference

Receiver-specific functions and structures.

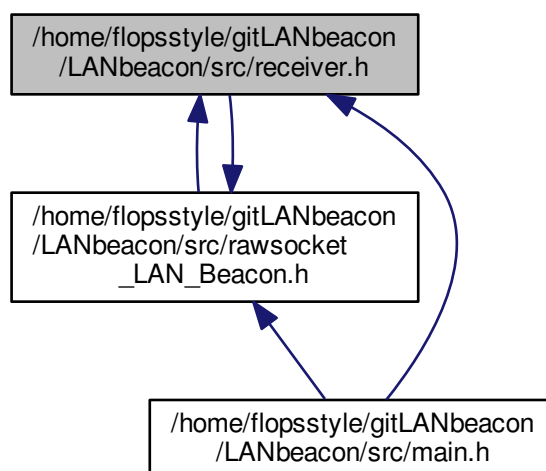
```
#include "define.h"
#include "openssl_sign.h"
```

```
#include "rawsocket_LAN_Beacon.h"
```

Include dependency graph for receiver.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [received\\_lan\\_beacon\\_frame](#)  
*Contains all the information related to one received frame.*
- struct [receiver\\_interfaces](#)  
*Contains all variables, that are needed to access sockets on interfaces.*
- struct [receiver\\_information](#)  
*Receiver configurations.*

## Functions

- char \*\* [evaluatelanbeacon](#) (struct [received\\_lan\\_beacon\\_frame](#) \*my\_received\_lan\_beacon\_frame, struct [open\\_ssl\\_keys](#) \*lanbeacon\_keys)  
*This function takes raw received LAN-Beacon frames and creates strings from them, that can be used for printing or further processing.*
- void [bananaPIprint](#) (struct [receiver\\_information](#) \*my\_receiver\_information)  
*This function prints the received content on the standard output and, if compiler flags are set, also on a C-Berry display.*

### 4.3.1 Detailed Description

Receiver-specific functions and structures.

#### Author

Dominik Bitzer

#### Date

2017

### 4.3.2 Function Documentation

#### 4.3.2.1 [bananaPIprint\(\)](#)

```
void bananaPIprint (
    struct receiver\_information * my_receiver_information )
```

This function prints the received content on the standard output and, if compiler flags are set, also on a C-Berry display.

#### Parameters

<a href="#">my_receiver_information</a>	receiver information struct, that contains display settings and contents that should be printed
---	---

#### 4.3.2.2 evaluatelanbeacon()

```
char** evaluatelanbeacon (
    struct received_lan_beacon_frame * my_received_lan_beacon_frame,
    struct open_ssl_keys * lanbeacon_keys )
```

This function takes raw received LAN-Beacon frames and creates strings from them, that can be used for printing or further processing.

##### Parameters

<i>my_received_lan_beacon_frame</i>	Pointer to one single received LAN-Beacon frame, that should be evaluated
<i>lanbeacon_keys</i>	Pointer to struct for keys, needed in order to verify authentication information

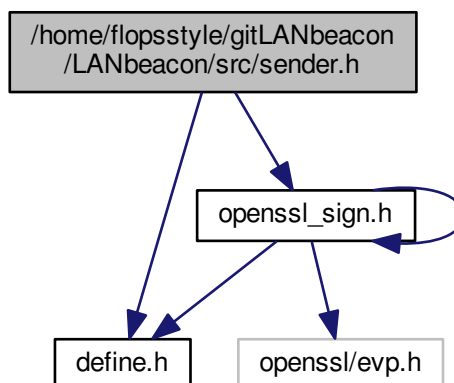
##### Returns

Returns parsed content as an array of TLV-descriptor and TLV-content pairs

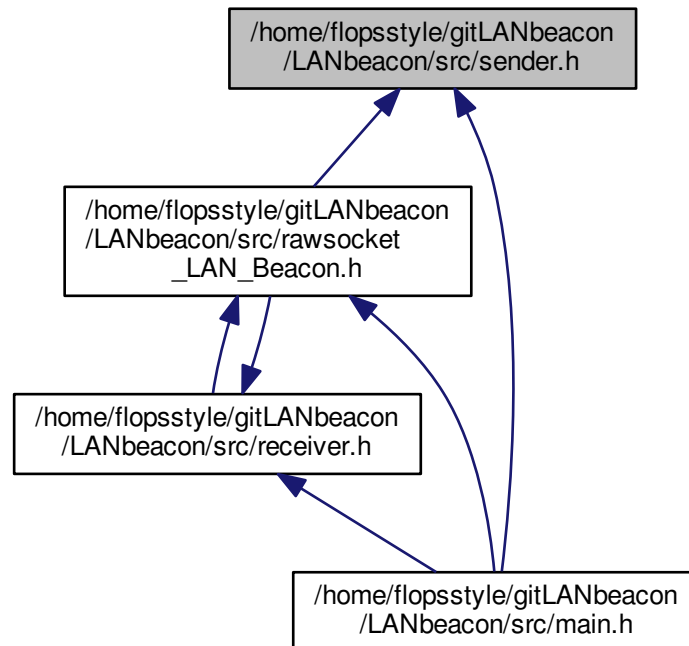
## 4.4 /home/flopsstyle/gitLANbeacon/LANbeacon/src/sender.h File Reference

Sender-specific functions and structures.

```
#include "define.h"
#include "openssl_sign.h"
Include dependency graph for sender.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [sender\\_information](#)  
*Sender configurations.*

## Functions

- char \* [mergedlanbeaconCreator](#) (int \*argc, char \*\*argv, struct [sender\\_information](#) \*my\_sender\_information)  
*Creates a LAN-Beacon PDU from the command line arguments.*
- void [transferToCombinedBeaconAndString](#) (unsigned char subtype, char \*TLVdescription, char \*\*combinedString, char \*source, char \*combinedBeacon, int \*currentByte)  
*Shortcut function for cases in which only a string is transferred, no binary format TLVs.*
- void [transferToCombinedBeacon](#) (unsigned char subtype, void \*source, char \*combinedBeacon, int \*currentByte, unsigned short int currentTLVlength)  
*Transferring the content of the field to the combined lanbeacon in binary format.*
- void [transferToCombinedString](#) (char \*TLVdescription, char \*\*combinedString, char \*source)  
*Transfer human-readable information to combined string.*
- void [ipParser](#) (int ip\_V4or6, char \*optarg, char \*\*combinedString, char \*combinedBeacon, int \*currentByte)  
*Parse IPv4 or IPv6 subnets to binary format.*



### 4.4.1 Detailed Description

Sender-specific functions and structures.

#### Author

Dominik Bitzer

#### Date

2017

### 4.4.2 Function Documentation

#### 4.4.2.1 ipParser()

```
void ipParser (
    int ip_V4or6,
    char * optarg,
    char ** combinedString,
    char * combinedBeacon,
    int * currentByte )
```

Parse IPv4 or IPv6 subnets to binary format.

Using regex to get IP-addresses from string input, then convert them to binary representation for transport

#### Parameters

<i>ip_V4or6</i>	Switch between IPv4 and IPv6 mode
<i>optarg</i>	String, which should be parsed
<i>combinedString</i>	Pointer to the string, that contains text representation of all contents
<i>combinedBeacon</i>	PDU of beacon, that TLVs should be added to
<i>currentByte</i>	current position in the Beacon-PDU

#### 4.4.2.2 mergedlanbeaconCreator()

```
char* mergedlanbeaconCreator (
    int * argc,
    char ** argv,
    struct sender_information * my_sender_information )
```

Creates a LAN-Beacon PDU from the command line arguments.

Howto for adding new fields:

1. Add defines for desired new field in [define.h](#)
2. Add desired options in [mergedlanbeaconCreator\(\)](#)

#### Parameters

<i>argc</i>	Number of arguments.
<i>argv</i>	Contents of arguments.

#### Returns

Returns an array, that contains the payload of a lanBeacon\_PDU

#### 4.4.2.3 transferToCombinedBeacon()

```
void transferToCombinedBeacon (
    unsigned char subtype,
    void * source,
    char * combinedBeacon,
    int * currentByte,
    unsigned short int currentTLVlength )
```

Transferring the content of the field to the combined lanbeacon in binary format.

#### Parameters

<i>subtype</i>	Subtype of the TLV
<i>source</i>	String contents, that should be included to the PDU
<i>combinedBeacon</i>	PDU of beacon, that TLVs should be added to
<i>currentByte</i>	current position in the Beacon-PDU
<i>currentTLVlength</i>	Length of the passed TLV

#### 4.4.2.4 transferToCombinedBeaconAndString()

```
void transferToCombinedBeaconAndString (
    unsigned char subtype,
    char * TLVdescription,
    char ** combinedString,
    char * source,
    char * combinedBeacon,
    int * currentByte )
```

Shortcut function for cases in which only a string is transferred, no binary format TLVs.

## Parameters

<i>subtype</i>	Subtype of the TLV
<i>TLVdescription</i>	Descriptor string of the TLV
<i>combinedString</i>	Pointer to the string, that contains text representation of all contents
<i>source</i>	String contents, that should be included to the PDU
<i>combinedBeacon</i>	PDU of beacon, that TLVs should be added to
<i>currentByte</i>	current position in the Beacon-PDU

## 4.4.2.5 transferToCombinedString()

```
void transferToCombinedString (
    char * TLVdescription,
    char ** combinedString,
    char * source )
```

Transfer human-readable information to combined string.

Transferring the content of the field to the combined string in human-readable format. If one combined string exceeds 507 byte limit of TLV it is put to the next combined string TLV

## Parameters

<i>TLVdescription</i>	Descriptor string of the TLV
<i>combinedString</i>	Pointer to the string, that contains text representation of all contents
<i>source</i>	String contents, that should be included to the PDU



# Index

/home/flopsstyle/gitLANbeacon/LANbeacon/src/define.h, [13](#)  
/home/flopsstyle/gitLANbeacon/LANbeacon/src/main.h, [15](#)  
/home/flopsstyle/gitLANbeacon/LANbeacon/src/receiver.h, [16](#)  
/home/flopsstyle/gitLANbeacon/LANbeacon/src/sender.h, [19](#)

authenticated\_mode  
    receiver\_information, [7](#)

bananaPIprint  
    receiver.h, [18](#)

challenge  
    received\_lan\_beacon\_frame, [6](#)  
current\_destination\_mac  
    received\_lan\_beacon\_frame, [6](#)  
current\_lan\_beacon\_pdu\_for\_printing  
    receiver\_information, [8](#)

evaluatelanbeacon  
    receiver.h, [19](#)

interface\_to\_send\_on  
    sender\_information, [10](#)

ipParser  
    sender.h, [21](#)

lan\_beacon\_ReceivedPayload  
    received\_lan\_beacon\_frame, [6](#)  
lan\_beacon\_pdu\_len  
    sender\_information, [10](#)  
lanBeacon\_PDU  
    sender\_information, [11](#)  
lanbeacon\_keys  
    receiver\_information, [8](#)  
    sender\_information, [10](#)

main  
    main.h, [16](#)  
main.h  
    main, [16](#)  
maxSockFd  
    receiver\_interfaces, [9](#)  
mergedlanbeaconCreator  
    sender.h, [21](#)  
my\_receiver\_interfaces  
    receiver\_information, [8](#)

numInterfaces  
    receiver\_interfaces, [9](#)  
number\_of\_currently\_received\_frames  
    receiver\_information, [8](#)  
open\_ssl\_keys, [5](#)  
parsedBeaconContents  
    received\_lan\_beacon\_frame, [6](#)  
payloadSize  
    received\_lan\_beacon\_frame, [6](#)  
pointers\_to\_received\_frames  
    receiver\_information, [8](#)

received\_lan\_beacon\_frame, [5](#)  
    challenge, [6](#)  
    current\_destination\_mac, [6](#)  
    lan\_beacon\_ReceivedPayload, [6](#)  
    parsedBeaconContents, [6](#)  
    payloadSize, [6](#)  
    successfullyAuthenticated, [6](#)  
    times\_left\_to\_display, [6](#)

receiver.h  
    bananaPIprint, [18](#)  
    evaluatelanbeacon, [19](#)  
receiver\_information, [7](#)  
    authenticated\_mode, [7](#)  
    current\_lan\_beacon\_pdu\_for\_printing, [8](#)  
    lanbeacon\_keys, [8](#)  
    my\_receiver\_interfaces, [8](#)  
    number\_of\_currently\_received\_frames, [8](#)  
    pointers\_to\_received\_frames, [8](#)  
    scroll\_speed, [8](#)  
receiver\_interfaces, [9](#)  
    maxSockFd, [9](#)  
    numInterfaces, [9](#)  
    sockfd, [9](#)  
    sockopt, [9](#)

scroll\_speed  
    receiver\_information, [8](#)  
send\_frequency  
    sender\_information, [11](#)  
sender.h  
    ipParser, [21](#)  
    mergedlanbeaconCreator, [21](#)  
    transferToCombinedBeacon, [22](#)  
    transferToCombinedBeaconAndString, [22](#)  
    transferToCombinedString, [23](#)  
sender\_information, [10](#)

- interface\_to\_send\_on, [10](#)
  - lan\_beacon\_pdu\_len, [10](#)
  - lanBeacon\_PDU, [11](#)
  - lanbeacon\_keys, [10](#)
  - send\_frequency, [11](#)
- sockfd
  - receiver\_interfaces, [9](#)
- sockopt
  - receiver\_interfaces, [9](#)
- successfullyAuthenticated
  - received\_lan\_beacon\_frame, [6](#)
- times\_left\_to\_display
  - received\_lan\_beacon\_frame, [6](#)
- transferToCombinedBeacon
  - sender.h, [22](#)
- transferToCombinedBeaconAndString
  - sender.h, [22](#)
- transferToCombinedString
  - sender.h, [23](#)