libnetfilter_log Reference Manual

x.y

Generated by Doxygen 1.4.6

Tue Mar 21 13:47:12 2006

Contents

1 libnetfilter_log File Index

1.1 libnetfilter_log File List

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

2 libnetfilter_log File Documentation

2.1 libnetfilter_log File Reference

2.1.1 Detailed Description

libnetfilter_log is a userspace library providing interface to packets that have been logged by the kernel packet filter. It is part of the netfilter project and can be found at http://www.netfilter.org/

Author:

libnetfilter_log (C) 2005 by Harald Welte <laforge@gnumonks.org>
This software may be used and distributed according to the terms of the GNU General Public License, incorporated herein by reference.
Documentation by Gregor Maier <gregor@majordomus.org>, with a lot of stuff taken from Brad Fisher's <brad@info-link.net> libnetfilter_queue documentation.

Typedefs

- typedef int nflog_callback (struct nflog_g_handle *gh, struct nfgenmsg *nfmsg, struct nflog_data *nfad, void *data)

  Callback prototype.

Functions

- nfnl_handle * nflog_nfnlh (struct nflog_handle *h)

  Returns the netfilter netlink handle used by h.

- nflog_handle * nflog_open (void)

  Obtains netfilter log connection handle.

- nflog_handle * nflog_open_nfnl (struct nfnl_handle *nfnlh)

  Obtains netfilter log connection handle.

- int nflog_close (struct nflog_handle *h)
  Close netfilter log connection.
- int nflog_bind_pf (struct nflog_handle *h, u_int16_t pf)
  Binds the netfilter_log kernel logging backend to PF.
- int nflog_unbind_pf (struct nflog_handle *h, u_int16_t pf)
  Unbinds the netfilter_log kernel logging backend from PF. (Dangerous).
- nflog_g_handle * nflog_bind_group (struct nflog_handle *h, u_int16_t num)
  Bind the connection handle to a loggroup.
- int nflog_unbind_group (struct nflog_g_handle *gh)
  Unbind and free a group handle.
- int nflog_set_mode (struct nflog_g_handle *gh, u_int8_t mode, unsigned int len)
  Sets the amount of data copied to userspace for each packet.
- int nflog_set_timeout (struct nflog_g_handle *gh, u_int32_t timeout)
  Set flush timeout.
- int nflog_set_flags (struct nflog_g_handle *gh, u_int16_t flags)
  UNUSED.
- int nflog_set_qthresh (struct nflog_g_handle *gh, u_int32_t qthresh)
  Set transmit threshold.
- int nflog_set_nlbufsiz (struct nflog_g_handle *gh, u_int32_t nlbufsiz)
  Set the kernel buffer size.
- int nflog_callback_register (struct nflog_g_handle *gh, nflog_callback *cb, void *data)
  Register callback function that receives packets.
- int nflog_fd (struct nflog_handle *h)
  Return fd associated with netfilter log handle.
- int nflog_handle_packet (struct nflog_handle *h, char *buf, int len)
  Handle packet(s) received from kernel.
- nfulnl_msg_packet_hdr * nflog_get_msg_packet_hdr (struct nflog_data *nfad)
  Returns netfilter log packet header.
- u_int32_t nflog_get_nfmark (struct nflog_data *nfad)
  Returns netfilter mark currently assigned to the packet.
- int nflog_get_timestamp (struct nflog_data *nfad, struct timeval *tv)
  Returns the timestamp of the packet.
- u_int32_t nflog_get_indev (struct nflog_data *nfad)

Return ingresss interface index.

- u_int32_t nflog_get_physindev (struct nflog_data *nfad)
  Return physical ingresss interface index.
- u_int32_t nflog_get_outdev (struct nflog_data *nfad)
  Return egress interface index.
- u_int32_t nflog_get_physoutdev (struct nflog_data *nfad)
  Return physical egress interface index.
- nfulnl_msg_packet_hw * nflog_get_packet_hw (struct nflog_data *nfad)
  Returns hardware addresss of the packet.
- int nflog_get_payload (struct nflog_data *nfad, char **data)
  Retrieve packet payload.
- char * nflog_get_prefix (struct nflog_data *nfad)
  Return log prefix.
- int nflog_get_uid (struct nflog_data *nfad, u_int32_t *uid)
  Return uid of packet "owner".
- int nflog_get_seq (struct nflog_data *nfad, u_int32_t *seq)
  UNUSED.
- int nflog_get_seq_global (struct nflog_data *nfad, u_int32_t *seq)
  UNUSED.

Variables

- int **nflog_errno**

2.1.2 Typedef Documentation

2.1.2.1 typedef int nflog_callback(struct nflog_g_handle *gh, struct nfgenmsg *nfmsg, struct nflog_data *nfad, void *data)

Callback prototype.

nflog_callback function pointeres are registered with libnetfilter_log to handle received packets. The callback function is called for each received packet.

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

nfmsg Pointer to a netfilter netlink message ???

nfd Data structure containing the netfilter netlink attributes received from kernel. This pointer is passed to many of the informational functions, such as nflog_get_nfmark(), nflog_get_payload(), etc. Should not be manipulated directly

data The value passed as data parameter of nflog_callback_register()

Returns:
You must return zero on success. nflog_handle_packet() will return this value.

2.1.3 Function Documentation

2.1.3.1 struct nfnl_handle* nflog_nfnlh (struct nflog_handle * *h*)

Returns the netfilter netlink handle used by h.

Parameters:
h A netfilter log handle. See nflog_open()

Returns:
The netfilter netlink handle of h, that is used by libnfnetlink to talk to the kernel.

2.1.3.2 struct nflog_handle* nflog_open (void)

Obtains netfilter log connection handle.

Obtains netfilter log connection handle. When you are finished with the handle, you should destroy it by calling nflog_close(). A new netlink connection is obtained internally and associated with the queue connection handle returned.

Returns:
Pointer to a new log handle or NULL on failure

2.1.3.3 struct nflog_handle* nflog_open_nfnl (struct nfnl_handle * *nfnlh*)

Obtains netfilter log connection handle.

Obtains netfilter log connection handle using an existing netlink connection. This function is used internally to implement nflog_open(), and should typically not be called directly.

Parameters:
nfnlh Pointer to netfilter netlink handler

Returns:
Pointer to a new log handle or NULL on failure

2.1.3.4 int nflog_close (struct nflog_handle * *h*)

Close netfilter log connection.

Close a netfilter log connection associated with the connection handle and free resources.

Parameters:
h Pointer to netfilter log connection handle as obtained from nflog_open()

Returns:
0 on successs, non-zero on error and errno is set accordingly

2.1.3.5 int nflog_bind_pf (struct nflog_handle * *h*, u_int16_t *pf*)

Binds the netfilter_log kernel logging backend to PF.

Binds the netfilter_log kernel logging bind to the specified protocol family (like PF_INET).

Parameters:

h Pointer to netfilter log connection handle.

pf Protocal Family (see socket(2))

Returns:

0 on successs, non-zero on error and errno is set accordingly
EEXIST libnetfilter_log handler is already registered for this protocol family. This error can be igrnored safely.
EBUSY Another libnetfilter_log handler is registered XXX:
EINVAL Invalid Protocol Family has been specified.

NOTE: libnfnetlink currently catches the error and issues a perror() call, which will clear errno. To get the behaviour described above, comment out the perror() call in libnfnetlink/src/libnfnetlink.c

2.1.3.6 int nflog_unbind_pf (struct nflog_handle * *h*, u_int16_t *pf*)

Unbinds the netfilter_log kernel logging backend from PF. (Dangerous).

Unbinds the netfilter_log kernel logging backend from PF. (Dangerous). Calling this function may break other programs using libnetfilter_log. You should not use it! Not to unbind the logger is safe!

Parameters:

h Pointer to netfilter log connection handle.

pf Protocal Family (see socket(2))

Returns:

0 on successs, non-zero on error and errno is set accordingly
EINVAL Invalid Protocol Family has been specified.

2.1.3.7 struct nflog_g_handle* nflog_bind_group (struct nflog_handle * *h*, u_int16_t *num*)

Bind the connection handle to a loggroup.

Bind the connection handle to a loggroup. This call returns a group handle, which is usesd for later calls and for identifying the loggroup.

Parameters:

h Pointer to netfilter log connection handle.

num The number of loggroup.

Returns:

gh A netfilter log group handle.

2.1.3.8 int nflog_unbind_group (struct nflog_g_handle * *gh*)

Unbind and free a group handle.

Unbinds the specified group handle from its loggroup. The gh is freed by this call.

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

Returns:

0 on successs, non-zero on error and errno is set accordingly?

2.1.3.9 int nflog_set_mode (struct nflog_g_handle * *gh*, u_int8_t *mode*, unsigned int *len*)

Sets the amount of data copied to userspace for each packet.

Sets the amount of data copied to userspace for each packet.

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

mode How much data to copy:

NFULNL_COPY_NONE Do not copy data to userspace

NFULNL_COPY_META Copy only packet queuing metadata (The actual packet data itself is not copied)

NFULNL_COPY_PACKET Copy metadata and up to range bytes of packet data

len The amount of packet data to copy when mode is NFULNL_COPY_PACKET. Use 0 or 0xffff to copy the whole packet.

Returns:

0 on success, non-zero on error. errno is set accordingly.

2.1.3.10 int nflog_set_timeout (struct nflog_g_handle * *gh*, u_int32_t *timeout*)

Set flush timeout.

Time before flushing queued packets from kernel to userspace. This prevents packets from never being flushed to userspace if qthreshhold is not request. Packets are either flushed to userspace when the timeout expires or when qthreshold packets are queued. (See nflog_set_qthresh()).

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

timeout The timeout to set in 1/100 s. (i.e. multiples of 10ms)??

Returns:

0 on success, non-zero on error. errno is set accordingly XXX: is this true??

2.1.3.11 int nflog_set_qthresh (struct nflog_g_handle * *gh*, u_int32_t *qthresh*)

Set transmit threshold.

Number of packet to queue inside kernel. Setting this value to, e.g. 10 accumulates ten packets inside the kernel and transmits them as one netlink multipart message to userspace. Default is 1 (for backwards compatibility). See also nflog_set_timeout().

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

qthresh The value of the threshold

Returns:

0 on success, non-zero on error. errno is set accordingly XXX: is this true??

2.1.3.12 int nflog_set_nlbufsiz (struct nflog_g_handle * *gh*, u_int32_t *nlbufsiz*)

Set the kernel buffer size.

Set the size of the kernel buffer that are used to packets before they are trasnmitted to userspace. Please note that the netlink message headers are also included, so you have to add this margin to the payload size you want to queue. ???

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

nlbufsiz The size of the kernel buffer in bytes.

Returns:

0 on success, non-zero on error. errno is set accordingly XXX: is this true?

2.1.3.13 int nflog_callback_register (struct nflog_g_handle * *gh*, nflog_callback * *cb*, void * *data*)

Register callback function that receives packets.

Register a callback function that receives packets for the given group. This callback will be called for each packet logged to userspace.

Parameters:

gh A netfilter log group handle (see nflog_bind_group())

cb Callback function pointer

data Custom data that is passed to the callback function

Returns:

0 on success, Won't fail.

2.1.3.14 int nflog_fd (struct nflog_handle * *h*)

Return fd associated with netfilter log handle.

Returns the filedescriptor associated with the netfilter log handle. This fd is used to received logged packets from the kernel using recv(2). The data received from recv is then passed ti nflog_handle_packet()

Parameters:

h Netfilter log handle. See nflog_open()

Returns:

Returns a file descriptor that can be used with the recv(2) system call.

2.1.3.15 int nflog_handle_packet (struct nflog_handle * *h*, char * *buf*, int *len*)

Handle packet(s) received from kernel.

Packets that have been received from the kernel with recv(2) are handed to nflog_handle_packet(), which will trigger the associated callback functions (see nflog_callback_register()).

Parameters:

h Netfilter log handle. See nflog_open()

buf The buffer containing the received data. This is the same buffer that recv(2) has filled.

len Size of the buffer. This is the length returned from the recv(2) call.

Returns:

0 on success, non-zero on error. Negative values are returned when internal errors occured. Otherwise the return value of the callback is returned (which must return zero on success).

2.1.3.16 struct nfulnl_msg_packet_hdr* nflog_get_msg_packet_hdr (struct nflog_data * *nfad*)

Returns netfilter log packet header.

Returns netfilter log packet header for the given nflog_data argument. This function is intended to be called from the callback.

Parameters:

nfad Netlink packet attribute data passed to callback function.

Returns:

Netlink packet header for the given packet data. The struct contains the following fields:
"u_int16_t hw_protocol" The ethertype of the packet in network byte order
"u_int8_t hook" The netfilter hook where this packet was intercepted

2.1.3.17 u_int32_t nflog_get_nfmark (struct nflog_data * *nfad*)

Returns netfilter mark currently assigned to the packet.

Parameters:

nfad Netlink packet attribute data passed to callback function.

Returns:

The netfilter mark currently assigned to the packet.

2.1.3.18 int nflog_get_timestamp (struct nflog_data * *nfad*, struct timeval * *tv*)

Returns the timestamp of the packet.

Parameters:

nfad Netlink packet attribute data passed to callback function.

tv Structure to fill with timestamp info

Returns:

0 on success, -1 on failure (tv is unmodified on failure).

2.1.3.19 u_int32_t nflog_get_indev (struct nflog_data * *nfad*)

Return ingresss interface index.

Parameters:
nfad Netlink packet attribute data passed to callback function.

Returns:
The index of the physical device the queued packet was received via. If the returned index is 0, the packet was locally generated or the physical input interface is no longer known. If the packet arrived on a bridged port, indev is the bridge group??

2.1.3.20 u_int32_t nflog_get_physindev (struct nflog_data * *nfad*)

Return physical ingresss interface index.

Parameters:
nfad Netlink packet attribute data passed to callback function.

Returns:
The index of the physical device the queued packet was received via. If the returned index is 0, the packet was locally generated or the physical input interface is no longer known. If the packet arrived on a bridged port physindev returns the number of the interface where the packet entered the system.

2.1.3.21 u_int32_t nflog_get_outdev (struct nflog_data * *nfad*)

Return egress interface index.

Parameters:
nfad Netlink packet attribute data passed to callback function.

See also:
nflog_get_indev()

2.1.3.22 u_int32_t nflog_get_physoutdev (struct nflog_data * *nfad*)

Return physical egress interface index.

Parameters:
nfad Netlink packet attribute data passed to callback function.

See also:
nflog_get_indev()

2.1.3.23 struct nfulnl_msg_packet_hw* nflog_get_packet_hw (struct nflog_data * *nfad*)

Returns hardware addresss of the packet.

Retrieves the hardware address associated with the given queued packet. For ethernet packets, the hardware address returned (if any) will be the MAC address of the packet source host. Empty for packets generated on the local machine

Parameters:
nfad Netlink packet attribute data passed to callback function.

Returns:
The source hardware address associated with the queued packet, or NULL if unknown. The struct contains the following fields:
"u_int16_t hw_addrlen" Length of the hardware address
"u_int8_t hw_addr[8]" The hardware address.

2.1.3.24 int nflog_get_payload (struct nflog_data * *nfad*, char ** *data*)

Retrieve packet payload.

Retrieve the packet data of logged packet, starting with the Layer 3 header (e.g. IP or IPv6). The actual amount of data available depends on the copy mode specified by nflog_set_mode().

Parameters:
nfad Netlink packet attribute data passed to callback function.
data A pointer to the packet data is returned in data.

Returns:
Length of the packet data pointed to by *data on success, -1 on failure.

2.1.3.25 char* nflog_get_prefix (struct nflog_data * *nfad*)

Return log prefix.

Parameters:
nfad Netlink packet attribute data passed to callback function.

Returns:
The Log Prefix of this packet.

2.1.3.26 int nflog_get_uid (struct nflog_data * *nfad*, u_int32_t * *uid*)

Return uid of packet "owner".

Parameters:
nfad Netlink packet attribute data passed to callback function
uid The uid of the packet owner is placed here.

Returns:
0 on success, -1 on error

Index