Phone Firewall Reference Manual

v0.01

Generated by Doxygen 1.5.4

Wed Jul 9 11:00:11 2008

Contents

Chapter 1

Phone Firewall Data Structure Index

1.1 Phone Firewall Data Structures

Here are the data structures with brief descriptions:

Chapter 2

Phone Firewall File Index

2.1 Phone Firewall File List

Here is a list of all files with brief descriptions:

Chapter 3

Phone Firewall Data Structure Documentation

3.1 Entry Struct Reference

`#include <libphonefirewall.h>`

Data Fields

- int country_code
- int area_code
- unsigned long long number
- char * name
- char * reason
- int priority

3.1.1 Detailed Description

Definition at line 52 of file libphonefirewall.h.

3.1.2 Field Documentation

3.1.2.1 int Entry::country_code

Definition at line 53 of file libphonefirewall.h.

Referenced by check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), and get_blacklist_entry_by_name().

3.1.2.2 int Entry::area_code

Definition at line 54 of file libphonefirewall.h.

Referenced by check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), and get_blacklist_entry_by_name().

3.1.2.3 unsigned long long Entry::number

Definition at line 55 of file libphonefirewall.h.

Referenced by check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), and get_blacklist_entry_by_name().

3.1.2.4 char* Entry::name

Definition at line 56 of file libphonefirewall.h.

Referenced by find_entry_by_name().

3.1.2.5 char* Entry::reason

Definition at line 57 of file libphonefirewall.h.

Referenced by find_entry_by_name(), and get_blacklist_entry_by_name().

3.1.2.6 int Entry::priority

Definition at line 58 of file libphonefirewall.h.

Referenced by check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), and find_entry_by_name().

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

- libphonefirewall.h

3.2 entry Struct Reference

Includes all informations for an entry.

`#include <libphonefirewall.h>`

3.2.1 Detailed Description

Includes all informations for an entry.

The struct which includes all information about entries (black- and whitelist).

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

- libphonefirewall.h

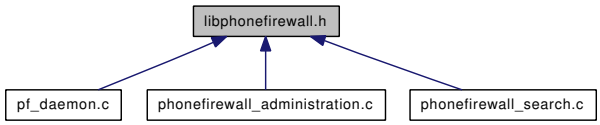
Chapter 4

Phone Firewall File Documentation

4.1 libphonefirewall.h File Reference

API of the phone firewall.

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



Data Structures

- struct Entry

Defines

- #define PRIO_ALL -999
- #define DB_FILE "db/phone-firewall.db"
- #define STMT_SIZE 1024
- #define MAX_LINE_LENGTH 512
- #define TB_COUNTRYCODE "countrycode"
- #define TB_AREACODE "areacode"
- #define TB_NUMBER "number"
- #define TB_NAME "name"
- #define TB_REASON "reason"
- #define TB_PRIORITY "priority"

Functions

- int add_blacklist_entry (int country_code, int area_code, unsigned long long number, char ∗name, char ∗reason, int priority)
- int rm_blacklist_entry (int country_code, int area_code, unsigned long long number)

- int check_blacklist_entry (int country_code, int area_code, unsigned long long number, int priority)
- int add_whitelist_entry (int country_code, int area_code, unsigned long long number, char *name, char *reason, int priority)
- struct Entry * get_blacklist_entry_by_name (char *name)
- struct Entry * get_blacklist_entry_by_number (int country_code, int area_code, unsigned long long number)
- int rm_whitelist_entry (int country_code, int area_code, unsigned long long number)
- int check_whitelist_entry (int country_code, int area_code, unsigned long long number, int priority)
- struct Entry * get_whitelist_entry_by_name (char *name)
- struct Entry * get_whitelist_entry_by_number (int country_code, int area_code, unsigned long long number)

Variables

- struct Entry entry

4.1.1 Detailed Description

API of the phone firewall.

Author:

Alex Oberhauser

The header file of the Phone Firewall. Blocks or accepts incoming phone calls, so it's possible to prevent disturbing phone calls. Provides a API which can used by other application to build nice programs.

Definition in file libphonefirewall.h.

4.1.2 Define Documentation

4.1.2.1 #define DB_FILE "db/phone-firewall.db"

Definition at line 34 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.2 #define MAX_LINE_LENGTH 512

Definition at line 36 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.3 #define PRIO_ALL -999

Definition at line 33 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), and evaluate_stmt().

4.1.2.4 #define STMT_SIZE 1024

Definition at line 35 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.5 #define TB_AREACODE "areacode"

Definition at line 39 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.6 #define TB_COUNTRYCODE "countrycode"

Definition at line 38 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.7 #define TB_NAME "name"

Definition at line 41 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), find_entry_by_name(), and get_blacklist_entry_by_name().

4.1.2.8 #define TB_NUMBER "number"

Definition at line 40 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), get_blacklist_entry_by_name(), rm_blacklist_entry(), and rm_whitelist_entry().

4.1.2.9 #define TB_PRIORITY "priority"

Definition at line 43 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), evaluate_stmt(), find_entry_by_name(), and get_blacklist_entry_by_name().

4.1.2.10 #define TB_REASON "reason"

Definition at line 42 of file libphonefirewall.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), find_entry_by_name(), and get_blacklist_entry_by_name().

4.1.3 Function Documentation

4.1.3.1 int add_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*, char * *name*, char * *reason*, int *priority*)

Add a number to the blacklist. The number will be blocked after that.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

name The name of the person.

reason Why you have blocked this person.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.
The value "PRIO_ALL" stands for all priorities.

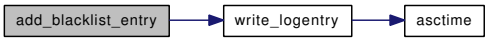
Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 75 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, PRIO_ALL, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.1.3.2 int add_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*, char * *name*, char * *reason*, int *priority*)

Add a number to the whitelist. The number will be accepted after that.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

name The name of the person.

reason Why you have blocked this person.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.
The value "PRIO_ALL" stands for all priorities.

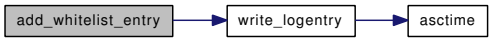
Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 117 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, PRIO_ALL, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.1.3.3 int check_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*, int *priority*)

Checks if a number is on the blacklist.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.
The value "PRIO_ALL" stands for all priorities.

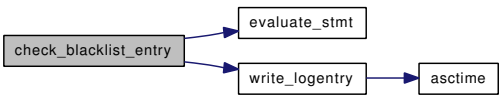
Returns:

If the number was found 1, otherwise 0.

Definition at line 234 of file phonefirewall_administration.c.

References Entry::area_code, Entry::country_code, DB_FILE, ERR_FLAG, evaluate_stmt(), INFO_FLAG, MAX_LINE_LENGTH, Entry::number, p_entry, Entry::priority, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, TB_PRIORITY, and write_logentry().

Here is the call graph for this function:



4.1.3.4 int check_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*, int *priority*)

Checks if a number is on the whitelist.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

The value "PRIO_ALL" stands for all priorities.

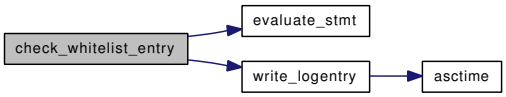
Returns:

If the number was found 1, otherwise 0.

Definition at line 293 of file phonefirewall_administration.c.

References Entry::area_code, Entry::country_code, DB_FILE, ERR_FLAG, evaluate_stmt(), INFO_FLAG, MAX_LINE_LENGTH, Entry::number, p_entry, Entry::priority, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, TB_PRIORITY, and write_logentry().

Here is the call graph for this function:



4.1.3.5 struct Entry* get_blacklist_entry_by_name (char * *name*) [read]

Search a entrie by name.

Parameters:

name The name of the person which is blocked.

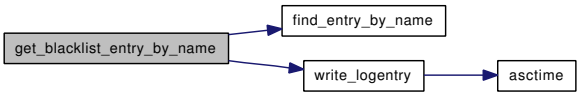
Returns:

entry Returns the found entry.

Definition at line 66 of file phonefirewall_search.c.

References Entry::area_code, ASCII_PERCENT_CHAR, Entry::country_code, DB_FILE, entry_array, ERR_FLAG, find_entry_by_name(), MAX_LINE_LENGTH, Entry::number, Entry::reason, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.1.3.6 struct Entry* get_blacklist_entry_by_number (int *country_code*, int *area_code*, unsigned long long *number*) [read]

Search a entrie by number (country code + area code + number).

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

Returns:

entry Returns the found entry.

Definition at line 115 of file phonefirewall_search.c.

4.1.3.7 struct Entry* get_whitelist_entry_by_name (char * *name*) [read]

Search a entrie by name.

Parameters:

name The name of the person which is accepted.

Returns:

entry Returns the found entry.

Definition at line 122 of file phonefirewall_search.c.

4.1.3.8 struct Entry* get_whitelist_entry_by_number (int *country_code*, int *area_code*, unsigned long long *number*) [read]

Search a entrie by number (country code + area code + number).

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

Returns:

entry Returns the found entry.

Definition at line 127 of file phonefirewall_search.c.

4.1.3.9 int rm_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*)

Removes a blocked number from the blacklist.

Parameters:

number The number which will be deleted.

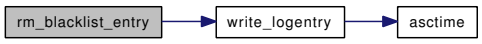
Returns:

If all goes right 0, otherwise an error code.

Definition at line 159 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, STMT_SIZE, TB_AREACODE, TB_-COUNTRYCODE, TB_NUMBER, and write_logentry().

Here is the call graph for this function:



4.1.3.10 int rm_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*)

Removes a accepted number from the whitelist.

Parameters:

number The number which will be deleted.

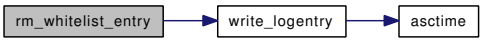
Returns:

If all goes right 0, otherwise an error code.

Definition at line 196 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, STMT_SIZE, TB_AREACODE, TB_-COUNTRYCODE, TB_NUMBER, and write_logentry().

Here is the call graph for this function:



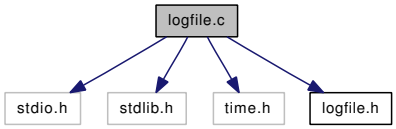
4.1.4 Variable Documentation

4.1.4.1 struct Entry entry

4.2 logfile.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "logfile.h"
```

Include dependency graph for logfile.c:



Functions

- char * asctime (const struct tm *timeptr)
- int write_logentry (char *msg, char *component, int flag)

4.2.1 Function Documentation

4.2.1.1 char* asctime (const struct tm * *timeptr*)

Compounds a humand readable date and time string.

Parameters:

timeptr A pointer to the actual time.

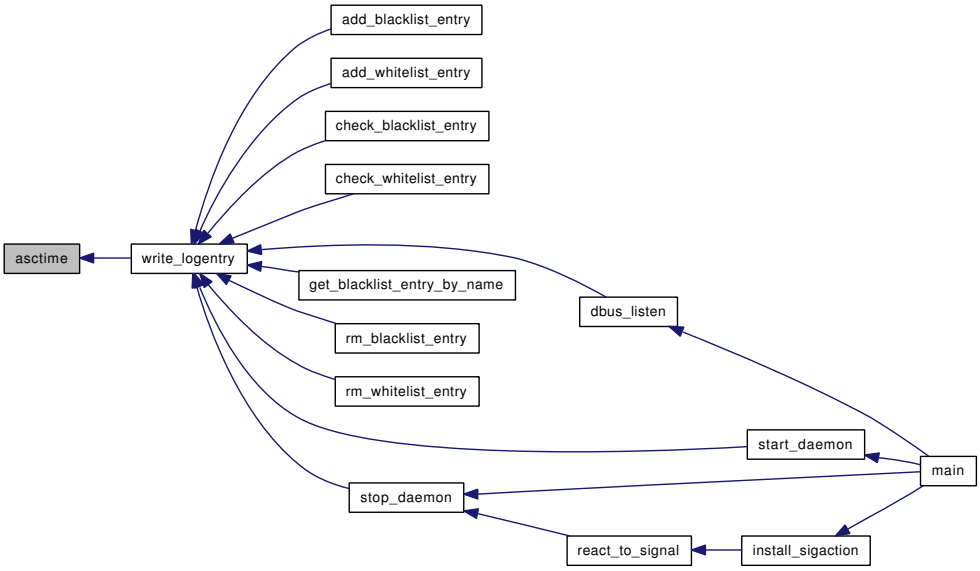
Returns:

The date and time as a string.

Definition at line 33 of file logfile.c.

Referenced by write_logentry().

Here is the caller graph for this function:



4.2.1.2 int write_logentry (char * *msg*, char * *component*, int *flag*)

Writes a logfile enty.

Parameters:

msg The message which should be written in the logfile.

component The program which calls the write_logentry function, for example "phonefirewall"

flag What message should be written. Use the defined flags.

Returns:

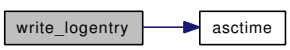
-1 if something fails, otherwise 0

Definition at line 56 of file logfile.c.

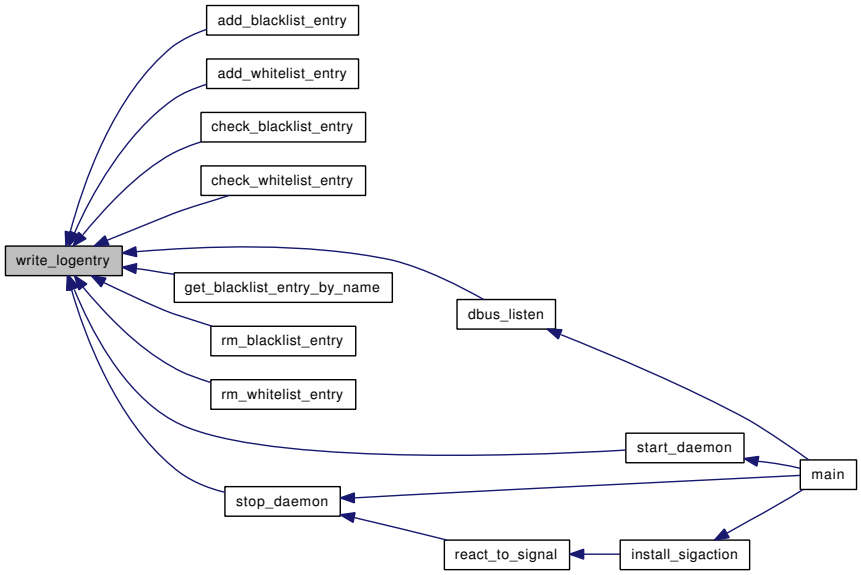
References asctime(), ERR_FLAG, INFO_FLAG, LOGFILE, MAX_ENTRY_LENGTH, UNKNOWN, and WARN_FLAG.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), dbus_listen(), get_blacklist_entry_by_name(), rm_blacklist_entry(), rm_whitelist_entry(), start_daemon(), and stop_daemon().

Here is the call graph for this function:

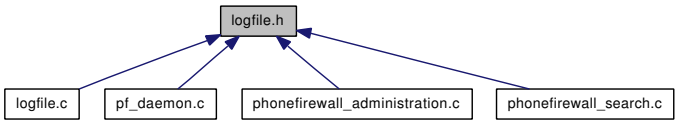


Here is the caller graph for this function:



4.3 logfile.h File Reference

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



Defines

- #define LOGFILE "log/moksec.log"
- #define MAX_ENTRY_LENGTH 128
- #define UNKNOWN 0
- #define ERR_FLAG 1
- #define WARN_FLAG 2
- #define INFO_FLAG 3

Functions

- int write_logentry (char *msg, char *component, int flag)

4.3.1 Define Documentation

4.3.1.1 #define ERR_FLAG 1

Definition at line 25 of file logfile.h.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), dbus_listen(), get_blacklist_entry_by_name(), rm_blacklist_entry(), rm_whitelist_entry(), and write_logentry().

4.3.1.2 #define INFO_FLAG 3

Definition at line 27 of file logfile.h.

Referenced by check_blacklist_entry(), check_whitelist_entry(), dbus_listen(), start_daemon(), stop_daemon(), and write_logentry().

4.3.1.3 #define LOGFILE "log/moksec.log"

Definition at line 21 of file logfile.h.

Referenced by write_logentry().

4.3.1.4 #define MAX_ENTRY_LENGTH 128

Definition at line 22 of file logfile.h.

Referenced by write_logentry().

4.3.1.5 #define UNKNOWN 0

Definition at line 24 of file logfile.h.

Referenced by write_logentry().

4.3.1.6 #define WARN_FLAG 2

Definition at line 26 of file logfile.h.

Referenced by write_logentry().

4.3.2 Function Documentation

4.3.2.1 int write_logentry (char * *msg*, char * *component*, int *flag*)

Writes a logfile enty.

Parameters:

msg The message which should be written in the logfile.

component The program which calls the write_logentry function, for example "phonefirewall"

flag What message should be written. Use the defined flags.

Returns:

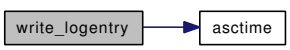
-1 if something fails, otherwise 0

Definition at line 56 of file logfile.c.

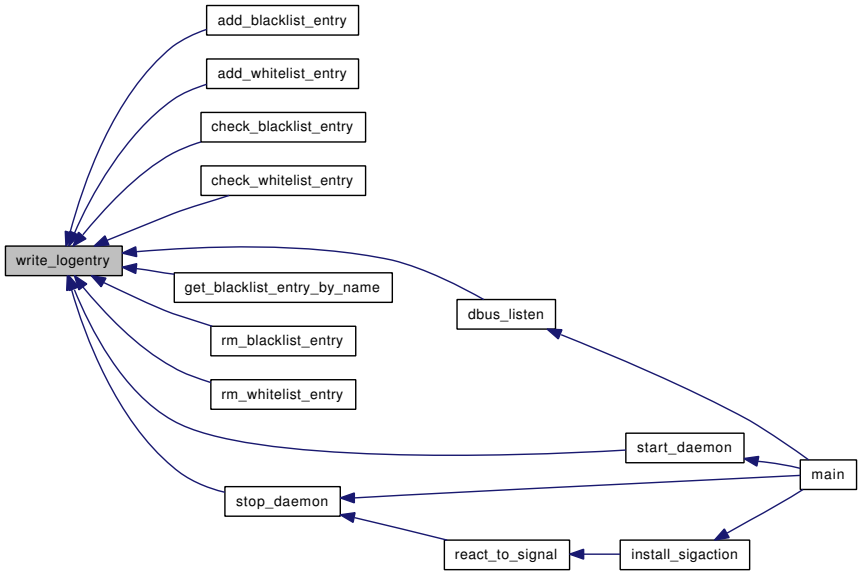
References asctime(), ERR_FLAG, INFO_FLAG, LOGFILE, MAX_ENTRY_LENGTH, UNKNOWN, and WARN_FLAG.

Referenced by add_blacklist_entry(), add_whitelist_entry(), check_blacklist_entry(), check_whitelist_entry(), dbus_listen(), get_blacklist_entry_by_name(), rm_blacklist_entry(), rm_whitelist_entry(), start_daemon(), and stop_daemon().

Here is the call graph for this function:



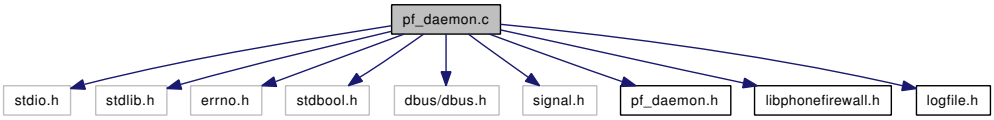
Here is the caller graph for this function:



4.4 pf_daemon.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <stdbool.h>
#include <dbus/dbus.h>
#include <signal.h>
#include "pf_daemon.h"
#include "libphonefirewall.h"
#include "logfile.h"
```

Include dependency graph for pf_daemon.c:



Functions

- void start_daemon ()
- void stop_daemon ()
- void dbus_listen ()
- void install_sigaction ()
- void react_to_signal (int sig)
- int main (int argc, char **argv)

4.4.1 Function Documentation

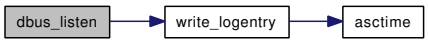
4.4.1.1 void dbus_listen ()

Definition at line 87 of file pf_daemon.c.

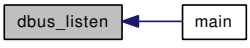
References ERR_FLAG, INFO_FLAG, and write_logentry().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



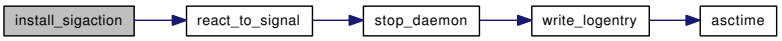
4.4.1.2 void install_sigaction ()

Definition at line 143 of file pf_daemon.c.

References react_to_signal().

Referenced by main().

Here is the call graph for this function:



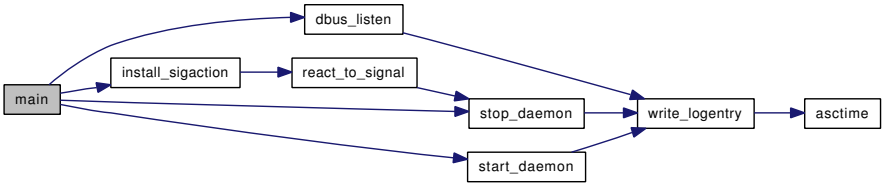
Here is the caller graph for this function:

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

Definition at line 38 of file pf_daemon.c.

References dbus_listen(), install_sigaction(), start_daemon(), and stop_daemon().

Here is the call graph for this function:



4.4.1.4 void react_to_signal (int *sig*)

Definition at line 157 of file pf_daemon.c.

References stop_daemon().

Referenced by install_sigaction().

Here is the call graph for this function:

Here is the caller graph for this function:

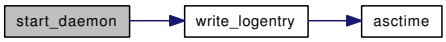
4.4.1.5 void start_daemon ()

Definition at line 53 of file pf_daemon.c.

References INFO_FLAG, LOCKFILE, and write_logentry().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

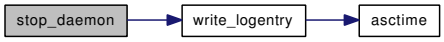
4.4.1.6 void stop_daemon ()

Definition at line 79 of file pf_daemon.c.

References INFO_FLAG, LOCKFILE, and write_logentry().

Referenced by main(), and react_to_signal().

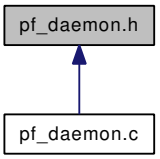
Here is the call graph for this function:



Here is the caller graph for this function:

4.5 pf_daemon.h File Reference

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



Defines

- #define LOCKFILE ".lock"

Functions

- void start_daemon ()
- void stop_daemon ()
- void dbus_listen ()

4.5.1 Define Documentation

4.5.1.1 #define LOCKFILE ".lock"

Definition at line 21 of file pf_daemon.h.

Referenced by start_daemon(), and stop_daemon().

4.5.2 Function Documentation

4.5.2.1 void dbus_listen ()

Function that provides method calls.

Bus name: to.networld.moksec.phonefirewall Object name: to.networld.moksec.phonefirewall.Object interface: to.networld.moksec.phonefirewall.Checking methods: checkblacklist(...) checkwhitelist(...)

4.5.2.2 void start_daemon ()

Starts the program as a daemon and creates a lockfile (specified in the LOCKFILE constant). So it's impossible to start the daemon twice.

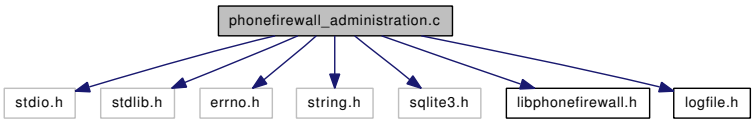
4.5.2.3 void stop_daemon ()

Stops the daemon and deletes the lockfile, so a new instance can be started.

4.6 phonefirewall_administration.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sqlite3.h>
#include "libphonefirewall.h"
#include "logfile.h"
```

Include dependency graph for phonefirewall_administration.c:



Functions

- int evaluate_stmt (sqlite3_stmt *pp_stmt, struct Entry *p_entry)
- int add_blacklist_entry (int country_code, int area_code, unsigned long long number, char *name, char *reason, int priority)
- int add_whitelist_entry (int country_code, int area_code, unsigned long long number, char *name, char *reason, int priority)
- int rm_blacklist_entry (int country_code, int area_code, unsigned long long number)
- int rm_whitelist_entry (int country_code, int area_code, unsigned long long number)
- int check_blacklist_entry (int country_code, int area_code, unsigned long long number, int priority)
- int check_whitelist_entry (int country_code, int area_code, unsigned long long number, int priority)

4.6.1 Function Documentation

4.6.1.1 int add_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*, char * *name*, char * *reason*, int *priority*)

Add a number to the blacklist. The number will be blocked after that.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

name The name of the person.

reason Why you have blocked this person.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

The value "PRIO_ALL" stands for all priorities.

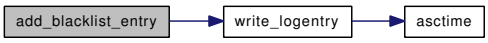
Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 75 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, PRIO_ALL, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.6.1.2 int add_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*, char * *name*, char * *reason*, int *priority*)

Add a number to the whitelist. The number will be accepted after that.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

name The name of the person.

reason Why you have blocked this person.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

The value "PRIO_ALL" stands for all priorities.

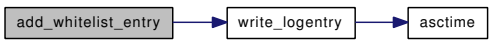
Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 117 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, PRIO_ALL, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.6.1.3 int check_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*, int *priority*)

Checks if a number is on the blacklist.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.
The value "PRIO_ALL" stands for all priorities.

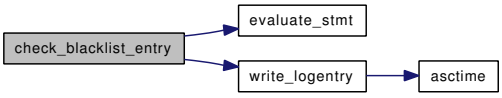
Returns:

If the number was found 1, otherwise 0.

Definition at line 234 of file phonefirewall_administration.c.

References Entry::area_code, Entry::country_code, DB_FILE, ERR_FLAG, evaluate_stmt(), INFO_FLAG, MAX_LINE_LENGTH, Entry::number, p_entry, Entry::priority, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, TB_PRIORITY, and write_logentry().

Here is the call graph for this function:



4.6.1.4 int check_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*, int *priority*)

Checks if a number is on the whitelist.

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

priority Gives the entry a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.
The value "PRIO_ALL" stands for all priorities.

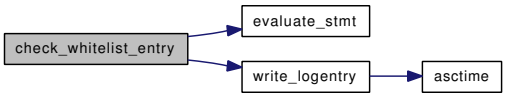
Returns:

If the number was found 1, otherwise 0.

Definition at line 293 of file phonefirewall_administration.c.

References Entry::area_code, Entry::country_code, DB_FILE, ERR_FLAG, evaluate_stmt(), INFO_FLAG, MAX_LINE_LENGTH, Entry::number, p_entry, Entry::priority, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, TB_PRIORITY, and write_logentry().

Here is the call graph for this function:



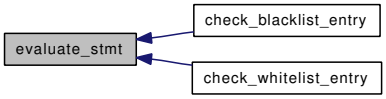
4.6.1.5 int evaluate_stmt (sqlite3_stmt * *pp_stmt*, struct Entry * *p_entry*)

Definition at line 28 of file phonefirewall_administration.c.

References Entry::area_code, Entry::country_code, Entry::number, PRIO_ALL, Entry::priority, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, and TB_PRIORITY.

Referenced by check_blacklist_entry(), and check_whitelist_entry().

Here is the caller graph for this function:



4.6.1.6 int rm_blacklist_entry (int *country_code*, int *area_code*, unsigned long long *number*)

Removes a blocked number from the blacklist.

Parameters:

number The number which will be deleted.

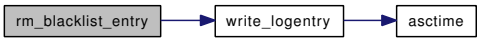
Returns:

If all goes right 0, otherwise an error code.

Definition at line 159 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, and write_logentry().

Here is the call graph for this function:



4.6.1.7 int rm_whitelist_entry (int *country_code*, int *area_code*, unsigned long long *number*)

Removes a accepted number from the whitelist.

Parameters:

number The number which will be deleted.

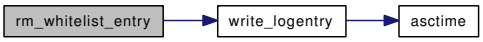
Returns:

If all goes right 0, otherwise an error code.

Definition at line 196 of file phonefirewall_administration.c.

References DB_FILE, ERR_FLAG, MAX_LINE_LENGTH, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NUMBER, and write_logentry().

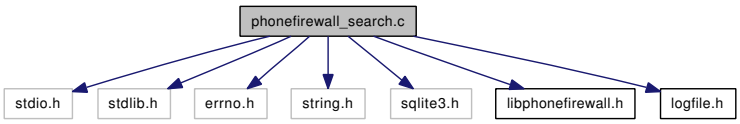
Here is the call graph for this function:



4.7 phonefirewall_search.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sqlite3.h>
#include "libphonefirewall.h"
#include "logfile.h"
```

Include dependency graph for phonefirewall_search.c:



Defines

- #define ASCII_PERCENT_CHAR 37
- #define MAX_ENTRY_ARRAY 1024

Functions

- struct Entry * find_entry_by_name (sqlite3_stmt *pp_stmt, char *name)
- struct Entry * get_blacklist_entry_by_name (char *name)
- struct Entry * get_blacklist_entry_by_number (int country_code, int area_code, unsigned long long number)
- struct Entry * get_whitelist_entry_by_name (char *name)
- struct Entry * get_whitelist_entry_by_number (int country_code, int area_code, unsigned long long number)

Variables

- struct Entry * p_entry = &entry
- struct Entry entry_array [MAX_ENTRY_ARRAY]

4.7.1 Define Documentation

4.7.1.1 #define ASCII_PERCENT_CHAR 37

Definition at line 28 of file phonefirewall_search.c.

Referenced by get_blacklist_entry_by_name().

4.7.1.2 #define MAX_ENTRY_ARRAY 1024

Definition at line 29 of file phonefirewall_search.c.

4.7.2 Function Documentation

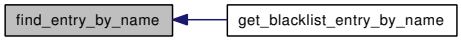
4.7.2.1 struct Entry* find_entry_by_name (sqlite3_stmt * *pp_stmt*, char * *name*) [read]

Definition at line 34 of file phonefirewall_search.c.

References Entry::area_code, Entry::country_code, Entry::name, Entry::number, Entry::priority, Entry::reason, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, and TB_REASON.

Referenced by get_blacklist_entry_by_name().

Here is the caller graph for this function:



4.7.2.2 struct Entry* get_blacklist_entry_by_name (char * *name*) [read]

Search a entrie by name.

Parameters:

name The name of the person which is blocked.

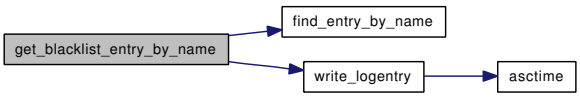
Returns:

entry Returns the found entry.

Definition at line 66 of file phonefirewall_search.c.

References Entry::area_code, ASCII_PERCENT_CHAR, Entry::country_code, DB_FILE, entry_array, ERR_FLAG, find_entry_by_name(), MAX_LINE_LENGTH, Entry::number, Entry::reason, STMT_SIZE, TB_AREACODE, TB_COUNTRYCODE, TB_NAME, TB_NUMBER, TB_PRIORITY, TB_REASON, and write_logentry().

Here is the call graph for this function:



4.7.2.3 struct Entry* get_blacklist_entry_by_number (int *country_code*, int *area_code*, unsigned long long *number*) [read]

Search a entrie by number (country code + area code + number).

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

Returns:

entry Returns the found entry.

Definition at line 115 of file phonefirewall_search.c.

4.7.2.4 struct Entry* get_whitelist_entry_by_name (char * *name*) [read]

Search a entrie by name.

Parameters:

name The name of the person which is accepted.

Returns:

entry Returns the found entry.

Definition at line 122 of file phonefirewall_search.c.

4.7.2.5 struct Entry* get_whitelist_entry_by_number (int *country_code*, int *area_code*, unsigned long long *number*) [read]

Search a entrie by number (country code + area code + number).

Parameters:

country_code The country code (for example 39 for Italy, 43 for Austria, and so one)

area_code The area code which indicates your mobile operator.

number The telephone number of the person (without country and area code.

Returns:

entry Returns the found entry.

Definition at line 127 of file phonefirewall_search.c.

4.7.3 Variable Documentation

4.7.3.1 struct Entry entry_array[MAX_ENTRY_ARRAY]

Definition at line 32 of file phonefirewall_search.c.

Referenced by get_blacklist_entry_by_name().

4.7.3.2 struct Entry* p_entry = &entry

Definition at line 31 of file phonefirewall_search.c.

Referenced by check_blacklist_entry(), and check_whitelist_entry().

Index