OP-WireStorm

Generated by Doxygen 1.13.2

1 Class Index 1

1.1 Class List 1

2 File Index 3

2.1 File List 3

3 Class Documentation 5

3.1 CTMPMessage Struct Reference 5

3.1.1 Detailed Description 5

4 File Documentation 7

4.1 wirestorm.cpp File Reference 7

4.1.1 Detailed Description 8

4.1.2 Function Documentation 8

4.1.2.1 calculate_checksum() 8

4.1.2.2 destination_listener_thread() 9

4.1.2.3 main() 9

4.1.2.4 message_forwarder_thread() 9

4.1.2.5 read_n_bytes() 9

4.1.2.6 source_listener_thread() 9

Index 11

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CTMPMessage
Structure to hold a complete CTMP message 5

Chapter 2

File Index

2.1 File List

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

wirestorm.cpp
CoreTech Message Protocol (CTMP) Proxy Server Implementation 7

Chapter 3

Class Documentation

3.1 CTMPMessage Struct Reference

Structure to hold a complete CTMP message.

Public Attributes

- std::vector< uint8_t > **data**

3.1.1 Detailed Description

Structure to hold a complete CTMP message.

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

- wirestorm.cpp

Chapter 4

File Documentation

4.1 wirestorm.cpp File Reference

CoreTech Message Protocol (CTMP) Proxy Server Implementation.

```
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <unistd.h>
#include <cerrno>
#include <chrono>
#include <condition_variable>
#include <cstring>
#include <iostream>
#include <mutex>
#include <queue>
#include <thread>
#include <vector>
```

Classes

- struct CTMPMessage

  Structure to hold a complete CTMP message.

Functions

- uint16_t calculate_checksum (uint8_t *header, uint8_t *data, uint16_t data_length)

  Calculate 16-bit one's complement checksum for CTMP messages.
- ssize_t read_n_bytes (int sockfd, uint8_t *buffer, size_t n_bytes, int timeout_sec)

  Reads a specific number of bytes from a socket with timeout.
- void source_listener_thread ()

  Thread function for listening to the single source client.
- void destination_listener_thread ()

  Thread function for listening to and accepting destination clients.
- void message_forwarder_thread ()

  Thread function for forwarding messages to destination clients.
- int main ()

  Main function - Entry point for the CTMP proxy server.

Variables

- const uint8_t **MAGIC_BYTE** = 0xCC
- const size_t **HEADER_SIZE** = 8
- const uint16_t **SOURCE_PORT** = 33333
- const uint16_t **DESTINATION_PORT** = 44444
- const uint16_t **MAX_PAYLOAD_SIZE** = 65535
- std::vector< int > **destination_clients**
- std::mutex **clients_mutex**
- std::queue< CTMPMessage > **message_queue**
- std::mutex **queue_mutex**
- std::condition_variable **queue_cv**

4.1.1 Detailed Description

CoreTech Message Protocol (CTMP) Proxy Server Implementation.

This file implements a multi-threaded proxy server that receives CTMP messages from a single source client and forwards them to multiple destination clients. The server validates message integrity for sensitive messages using checksums and handles client connections dynamically.

Author

Kieran Cookson

Date

16 August 2025

4.1.2 Function Documentation

4.1.2.1 calculate_checksum()

```
uint16_t calculate_checksum (
            uint8_t * header,
            uint8_t * data,
            uint16_t data_length)
```

Calculate 16-bit one's complement checksum for CTMP messages.

Implements the standard Internet checksum algorithm. The checksum field in the header is set to 0xCCCC during calculation.

Parameters

header	Pointer to the message header (8 bytes)
data	Pointer to the message payload data
data_length	Length of the payload data in bytes

Returns

16-bit one's complement checksum value

4.1.2.2 destination_listener_thread()

```
void destination_listener_thread ()
```

Thread function for listening to and accepting destination clients.

This function handles incoming connections from destination clients. Multiple destination clients can connect simultaneously.

4.1.2.3 main()

```
int main ()
```

Main function - Entry point for the CTMP proxy server.

Initializes and starts all server threads, then keeps the server running.

Returns

Exit status (0 for normal termination)

4.1.2.4 message_forwarder_thread()

```
void message_forwarder_thread ()
```

Thread function for forwarding messages to destination clients.

This function continuously processes messages from the queue and forwards them to all connected destination clients. Failed clients are automatically removed.

4.1.2.5 read_n_bytes()

```
ssize_t read_n_bytes (
            int sockfd,
            uint8_t * buffer,
            size_t n_bytes,
            int timeout_sec)
```

Reads a specific number of bytes from a socket with timeout.

This function ensures that exactly n_bytes are read from the socket, handling partial reads and network interruptions.

Parameters

sockfd	Socket file descriptor to read from
buffer	Buffer to store the read data
n_bytes	Number of bytes to read
timeout_sec	Timeout in seconds for the read operation

Returns

Number of bytes read on success, 0 if connection closed, -1 on error/timeout

4.1.2.6 source_listener_thread()

```
void source_listener_thread ()
```

Thread function for listening to the single source client.

This function handles incoming connections from source clients, validates CTMP messages, performs checksum verification for sensitive messages, and queues valid messages for forwarding. Only one source client is accepted at a time.

Index

calculate_checksum
    wirestorm.cpp, 8
CTMPMessage, 5

destination_listener_thread
    wirestorm.cpp, 8

main
    wirestorm.cpp, 9
message_forwarder_thread
    wirestorm.cpp, 9

read_n_bytes
    wirestorm.cpp, 9

source_listener_thread
    wirestorm.cpp, 9

wirestorm.cpp, 7
    calculate_checksum, 8
    destination_listener_thread, 8
    main, 9
    message_forwarder_thread, 9
    read_n_bytes, 9
    source_listener_thread, 9