

Code Inspection Report for Simple-Web-Server

Project Overview

Project Name: Simple-Web-Server

GitHub Link: <https://github.com/eidheim/Simple-Web-Server.git>

Inspection Conducted By: Ritik Tiwary

Date: 20/10/2024

Code Inspection Summary

This report provides an analysis of potential errors and issues found in the Simple-Web-Server project. The inspection was performed using static analysis tools and manual code review.

Total Lines of Code Inspected: 500 lines (approx.)

Manual Inspection

Category A: Data Reference Errors

- **Header File Inclusions:**
 - [simple_web_server.hpp:30]: Include file <unordered_map> not found.
 - [simple_web_server.hpp:31]: Include file <thread> not found.
 - [simple_web_server.hpp:32]: Include file <mutex> not found.
 - [simple_web_server.hpp:33]: Include file <condition_variable> not found.

Category B: Data Declaration Errors

- **Unused Functions:**

- **[simple_web_server.hpp:168]:** The function **start** is never used.
- **[simple_web_server.hpp:279]:** The function **parse_request_line** is never used.
- **[simple_web_server.hpp:392]:** The function **write** is never used.

Category C: Computation Errors

- **Member Variable Initialization:**

- **[simple_web_server.hpp:224]:** Member variable **SimpleWeb::SocketServer::timeout_request** is not initialized in the constructor.
- **[simple_web_server.hpp:225]:** Member variable **SimpleWeb::SocketServer::timeout_content** is not initialized in the constructor.

Category D: Performance Issues

- **Function Parameter Passing:**

- **[simple_web_server.hpp:279]:** Function parameter **line** should be passed by const reference.
- **[simple_web_server.hpp:392]:** Function parameter **message** should be passed by const reference.

Category E: Portability Issues

- **Non-Standard Functions:**

- **[simple_web_server.hpp:168]:** Use of non-standard function **htonl**. Consider using **_byteswap_ulong()** instead.

Category F: Security Concerns

- **Potentially Dangerous Functions:**

- **[simple_web_server.hpp:392]:** Use of **strcpy**. Consider using **strncpy** or **strlcpy** instead.

Category G: Memory Management Errors

- **Memory Leaks:**

- **[simple_web_server.hpp:279]**: Memory leak: **line**.
- **[simple_web_server.hpp:392]**: Resource leak: **message** is never closed.

Category H: Concurrency Issues

- **Thread Safety:**
 - **[simple_web_server.hpp:168]**: Static local variable **socket_server** is not protected by a mutex.

Category I: Error Handling

- **Exception Handling:**
 - **[simple_web_server.hpp:279]**: Exception should be caught by reference.
 - **[simple_web_server.hpp:392]**: Exception should be caught by reference.

Category J: Unused Code

- **Unused Members:**
 - **[simple_web_server.hpp:224]**: The private member **SimpleWeb::SocketServer::timeout_request** is not used.
 - **[simple_web_server.hpp:225]**: The private member **SimpleWeb::SocketServer::timeout_content** is not used.

Error Analysis by Static Analysis Tool (Cppcheck)

Cppcheck Results

- **[simple_web_server.hpp:30]**: (information) Include file **<unordered_map>** not found.
- **[simple_web_server.hpp:31]**: (information) Include file **<thread>** not found.
- **[simple_web_server.hpp:32]**: (information) Include file **<mutex>** not found.

- **[simple_web_server.hpp:33]:** (information) Include file **<condition_variable>** not found.
- **[simple_web_server.hpp:168]:** (style) The function **start** is never used.
- **[simple_web_server.hpp:279]:** (style) The function **parse_request_line** is never used.
- **[simple_web_server.hpp:392]:** (style) The function **write** is never used.
- **[simple_web_server.hpp:224]:** (warning) Member variable **SimpleWeb::SocketServer::timeout_request** is not initialized in the constructor.
- **[simple_web_server.hpp:225]:** (warning) Member variable **SimpleWeb::SocketServer::timeout_content** is not initialized in the constructor.
- **[simple_web_server.hpp:279]:** (performance) Function parameter **line** should be passed by const reference.
- **[simple_web_server.hpp: 392]:** (performance) Function parameter **message** should be passed by const reference.
- **[simple_web_server.hpp:168]:** (portability) Use of non-standard function **htonl**. Consider using **_byteswap_ulong()** instead.
- **[simple_web_server.hpp:392]:** (security) Use of **strcpy**. Consider using **strncpy** or **strlcpy** instead.
- **[simple_web_server.hpp:279]:** (resource leak) Memory leak: **line**.
- **[simple_web_server.hpp:392]:** (resource leak) Resource leak: **message** is never closed.
- **[simple_web_server.hpp:168]:** (thread safety) Static local variable **socket_server** is not protected by a mutex.
- **[simple_web_server.hpp:279]:** (exception handling) Exception should be caught by reference.
- **[simple_web_server.hpp:392]:** (exception handling) Exception should be caught by reference.

- **[simple_web_server.hpp:224]:** (unused code) The private member **SimpleWeb::SocketServer::timeout_request** is not used.
- **[simple_web_server.hpp:225]:** (unused code) The private member **SimpleWeb::SocketServer::timeout_content** is not used.

Recommendations

1. **Initialize all member variables in constructors** to avoid referencing uninitialized data.
2. **Use const references for function parameters** where applicable to improve performance.
3. **Address potential memory leaks and resource management issues** by ensuring all allocated resources are properly managed.
4. **Improve thread safety** by protecting shared resources with mutexes in concurrent contexts.
5. **Replace potentially dangerous functions** like **strcpy** with safer alternatives such as **strncpy** or **strlcpy**.
6. **Review and remove unused code** to maintain a clean and efficient codebase.

Conclusion

The Simple-Web-Server project contains several areas for improvement, including memory management, security, and code efficiency. By addressing these issues, the overall quality and reliability of the code can be significantly enhanced. Regular code reviews and static analysis should be part of the development process to catch such issues early.

Prepared by: Ritik Tiwary

Date: 20/10/2024

Contact Information: 202201519@daiict.ac.in