Coreon MCP Execution Engine

Smart Contract Audit

- Coreon MCP Execution Engine Audit Summary
- Coreon MCP Execution Engine Audit
  - Document information
    - Audit results
  - Vulnerability analysis
    - Vulnerability distribution
    - Summary of audit results
    - Audited target file

Coreon MCP Execution Engine Audit Summary

Project name : Coreon MCP Execution Engine

Code URL : https://github.com/CoreonMcp/Coreon-MCP-Execution-Engine/tree/audit-code

Commit : a7bfd3b389ef87b6679158e6bee1e8556e6f2a78

Project target : Coreon MCP Execution Engine Audit

Test result : PASSED

Audit Info

Audit NO : 0X202510030006

Audit Team : Armors Labs

Audit Proofreading: https://armors.io/#project-cases

Coreon MCP Execution Engine Audit

The Coreon MCP Execution Engine team asked us to review and audit their Coreon MCP Execution Engine. We looked at the code and now publish our results.

Here is our assessment and recommendations, in order of importance.

Document information

Name	Auditor	Version	Date
Coreon MCP Execution Engine Audit	Rock, Roy, Rico, Susan	1.0.0	2025-10-03

Audit results

Note that as of the date of publishing, the above review reflects the current understanding of known security patterns as they relate to the Coreon MCP Execution Engine The above should not be construed as investment advice.

Based on the widely recognized security status of the current underlying blockchain, this audit report is valid for 3 months from the date of output.

Disclaimer

Armors Labs Reports is not and should not be regarded as an "approval" or "disapproval" of any particular project or team. These reports are not and should not be regarded as indicators of the economy or value of any "product" or "asset" created by any team. Armors do not cover testing or auditing the integration with external contract or services

Armors Labs Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by blockchain technology. Armors does not guarantee the safety or functionality of the technology agreed to be analyzed.

Armors Labs

Armors Labs postulates that the information provided is not missing, tampered, deleted or hidden. If the information provided is missing, tampered, deleted, hidden or reflected in a way that is not consistent with the actual situation, Armors Labs shall not be responsible for the losses and adverse effects caused. Armors Labs Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

Vulnerability analysis

Vulnerability distribution

vulnerability level	number
Critical severity	0
High severity	0
Medium severity	0
Low severity	0

Summary of audit results

Vulnerability	status
Authentication and Authorization	safe
Input and Output Data Security	safe
Dependency and Third-Party Library Security	safe
Environment Variables and Key Management	safe
AI/LLM Interface Security	safe

Armors Labs

Audited target file

```
.
├── assets
│   ├── gif
│   │   ├── API Server Mode_2.gif
│   │   ├── Claude.gif
│   │   ├── Index.gif
│   │   ├── Start CLI Mode.gif
│   │   └── Telegram Bot Mode.gif
│   └── icon.png
├── bot
│   └── telegram_bot.py
├── cli
│   ├── __init__.py
│   └── mcp_cli.py
├── config
│   └── tools.yaml
├── core
│   ├── loader.py
│   └── router.py
├── main.py
├── README.md
```

```
├── server
│   └── mcp_server.py
├── service
│   ├── llm_provider.py
│   ├── llm.py
│   ├── planner.py
│   └── prompt.py
├── tools
│   ├── market
│   │   ├── __init__.py
│   │   ├── logic.py
│   │   └── schema.py
│   └── token
│       ├── __init__.py
│       ├── logic.py
│       └── schema.py
└── utils
    ├── common.py
    ├── kline_text_renderer.py
    ├── logging_utils.py
    └── step_executor.py

13 directories, 31 files
```

Armors Labs

armors.io

contact@armors.io