MOTECH AUDIT

SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

2025

FTE TOKEN AUDIT REPORT

25 JUNE 2025

SECURITY ASSESMENT
SUMMARY

This report has been prepared for FTE TOKEN Audit Report smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognise ed library. A comprehensive examination has been performed, utilising Static Analysis and ManualReview techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

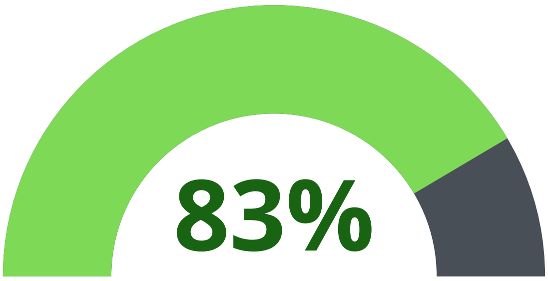
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

SECURITY ASSESMENT

SUMMARY

Pass/Fail	Analyzer	Description
	Motech Audit	Vulnerability analyzer built by Motech Audit that quickly searches within smart contracts for vulnerable code fragments not only exact but also syntactically different clones.
	Formal Verifier	Sound formal verifier built by Motech Audit that analyzes within smart contracts for proving that a program satisfies nonexistence of vulnerability such as integer overflow.
	Achilles	Symbolic analyzer built by Motech Audit that extracts code patterns within smart contracts for finding syntactic and semantic bugs and vulnerabilities based on SWC specification.

Pass   Compile Error   Failed



Contract Name FTE

Compiler Version v0.8.24+commit.e11b9ed9

License Type MIT

Contract Chain BSC Mainnet

(STATEMENT : MOTECH AUDIT ONLY ISSUES THIS REPORT BASED ON THE FACT THAT HAS OCCURRED OR EXISTED BEFORE THE REPORT IS ISSUED, AND BEARS THE CORRESPONDING RESPONSIBILITY IN THIS REGARD. FOR THE FACTS OCCUR OR EXIST LATER AFTER THE REPORT, MOTECH AUDIT CANNOT JUDGE THE SECURITY STATUS OF ITS SMART CONTRACT. MOTECH AUDIT IS NOT RESPONSIBLE FOR IT. THE SECURITY AUDIT ANALYSIS AND OTHER CONTENTS OF THIS REPORT ARE BASED ON THE DOCUMENTS AND MATERIALS PROVIDED BY THE INFORMATION PROVIDER TO MOTECH AUDIT AS OF THE DATE OF THIS REPORT (REFERRED TO AS "THE PROVIDED INFORMATION"). MOTECH AUDIT ASSUMES THAT: THERE HAS BEEN NO INFORMATION MISSING, TAMPERED, DELETED, OR CONCEALED. IF THE INFORMATION PROVIDED HAS BEEN MISSED, MODIFIED, DELETED, CONCEALED OR REFLECTED AND IS INCONSISTENT WITH THE ACTUAL SITUATION, MOTECH AUDIT WILL NOT BEAR ANY RESPONSIBILITY FOR THE RESULTING LOSS AND ADVERSE EFFECTS. MOTECH AUDIT WILL NOT BEAR ANY RESPONSIBILITY FOR THE BACKGROUND OR OTHER CIRCUMSTANCES OF THE PROJECT.)

MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

DISCLAIMER

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and MotechAudit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (MotechAudit) owe no duty of care towards you or any other person, nor does MotechAudit make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and MotechAudit hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, MotechAudit hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against MotechAudit, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

SECURITY ASSESMENT

BACKGROUND

Motech Audit was commissioned by my fusion international pte ltd to perform an audit of smart contracts:

Contract.sol
Contract Name **FTE**
Compiler Version **v0.8.24+commit.e11b9ed9**
License Type **MIT**
Contract Chain **BSC Mainnet**

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

SECURITY ASSESMENT

VULNERABILITY & RISK LEVEL

4 Total Issues

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Total Issues	Vulnerability	Risk (Required Action)
Critical	0	A vulnerability that can disrupt the contract functioning in a number of scenarios,or creates a risk that the contract may be broken	Immediate action to reduce risk level.
High	0	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	1	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Note	1	A vulnerability that have informational character but is not affecting any of the code.	An observation that does not determine a level of risk.

MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESMENT

AUDITING STRATEGY AND TECHNIQUES APPLIED

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

- Review of the specifications, sources, and instructions provided to Motech Audit to make sure we understand the size, scope, and functionality of the smart contract.
- Manual review of code, which is the process of reading source code line-byline in an attempt to identify potential vulnerabilities.
- Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Motech Audit describe.

2. Testing and automated analysis that includes the following:

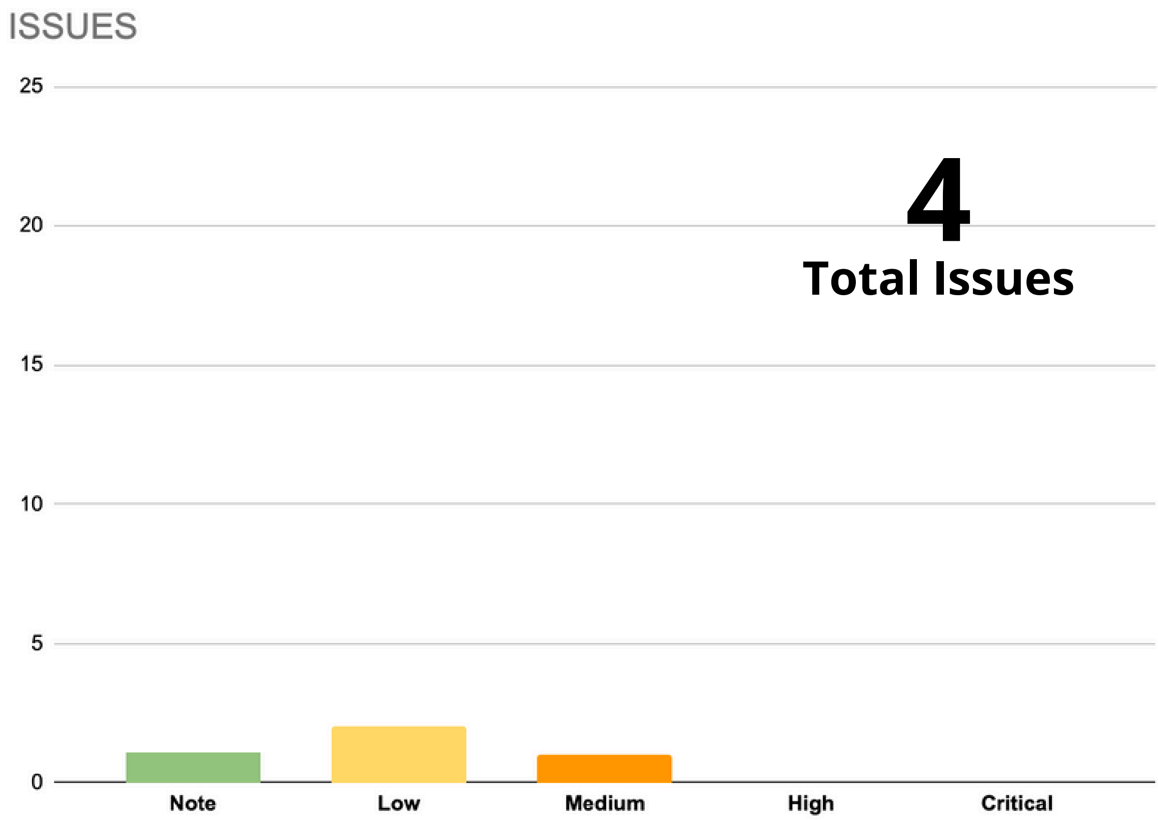
- Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
- Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

SECURITY ASSESMENT

FINDINGS



Audit Score Summary

Category	Score (Out of 10)
Code Correctness	9
Security	8.5
Upgrade Safety	8.5
Access Control	10
Gas Efficiency	9.5
Documentation	8.5
Test Coverage	N/A
Overall Audit Score	**83/100**

SECURITY ASSESMENT

PROJECT OVERVIEW

The Fusion Token Equity (FTE) contract implements a standard ERC-20 token with controlled minting and burning functionalities. It includes owner-based access control for protocol assignment and enforces a maximum cap of 125 million tokens. The audit was performed to assess code correctness, adherence to ERC-20 standards, security vulnerabilities, gas optimization, and upgrade readiness.

Key Features:

- **ERC20 Token using OpenZeppelin** – Standard-compliant, secure, and audited token base.
- **Total supply cap of 125 million tokens** – Enforced at the contract level via internal `mint()` restriction.
- **Mint and burn functionality restricted to FTXProtocol** – Only the designated protocol address can create or destroy tokens.
- **Ownership-based governance of protocol privileges** – The `owner` can configure the `FTXProtocol` address.
- **Minimalist design with no upgrade logic** – Reduced attack surface and simpler verification.
- **Constructor-based ownership** – Ensures deploy-time governance assignment.
- **No external dependencies beyond OpenZeppelin** – No third-party oracles, proxies, or delegate calls.

In-Scope Contracts

- FTS.sol: The main contract implementing the ERC20 token.

External Libraries Used:

- OpenZeppelin Contracts v5.3.0
  - ERC20
  - Context
  - Ownable
  - SafeMath (implicitly)

CODE

[Medium] Missing require(to != address(0)) in burn()

- **Location:** burn(address to, uint256 amount)
- **Impact:** Burning from address(0) can result in silent accounting errors.
- **Recommendation:** Add require(to != address(0), "Cannot burn from zero address").

[Low] Redundant check in setFTXProtocol

- **Location:** setFTXProtocol
- **Observation**: Redundant check require(address(account) != address(0)).
- **Impact**: Low
- **Recommendation**: Either enforce this check meaningfully or remove it and document that 0x0 is not allowed.

[Low] Lack of Events for setFTXProtocol

- **Observation:** Changing protocol address should emit an event for off-chain tracking

- **Recommendation**: Add event FTXProtocolUpdated(address newProtocol) and emit in setFTXProtocol

SECURITY ASSESMENT

CODE

[Info] Gas Optimization for Constants

- **Location:** Mint cap comparison
- **Observation:** 125000000 * 1e18 recalculated each time
- **Recommendation:** Store this as a constant (e.g., uint256 public constant MAX_SUPPLY = 125_000_000 * 1e18;)

Security Checks Performed

Check Type	Status
Reentrancy	✅ Passed
Integer Overflow/Underflow	✅ Passed (SafeMath by default in ^0.8.0)
Access Control	✅ Passed
Token Supply Cap	✅ Enforced in `mint()`
External Calls/Delegates	✅ Not used
Denial of Service	✅ Passed
Block Dependency	✅ None
tx.origin Usage	✅ None
Upgrade Safety	⚠ Not upgradeable
Front-running Resistance	✅ Passed

SECURITY ASSESMENT

CONCLUSION

Final Recommendations:

- Emit events for key changes (e.g., FTXProtocol updates).
- Add NatSpec comments for public/external functions.
- Consider renaming burn() to burnFrom() for clarity.
- Optionally cache cap value for gas efficiency.
- Ensure comprehensive test coverage for all roles.

The Fusion Token Equity (FTE) smart contract demonstrates a clean and minimal implementation of an ERC-20 token with additional mint/burn logic and access control. No critical or major issues were found. Minor improvements in event logging and safe programming practices are recommended to enhance maintainability and auditability.

Motech Audit note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.

PASSED

MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT