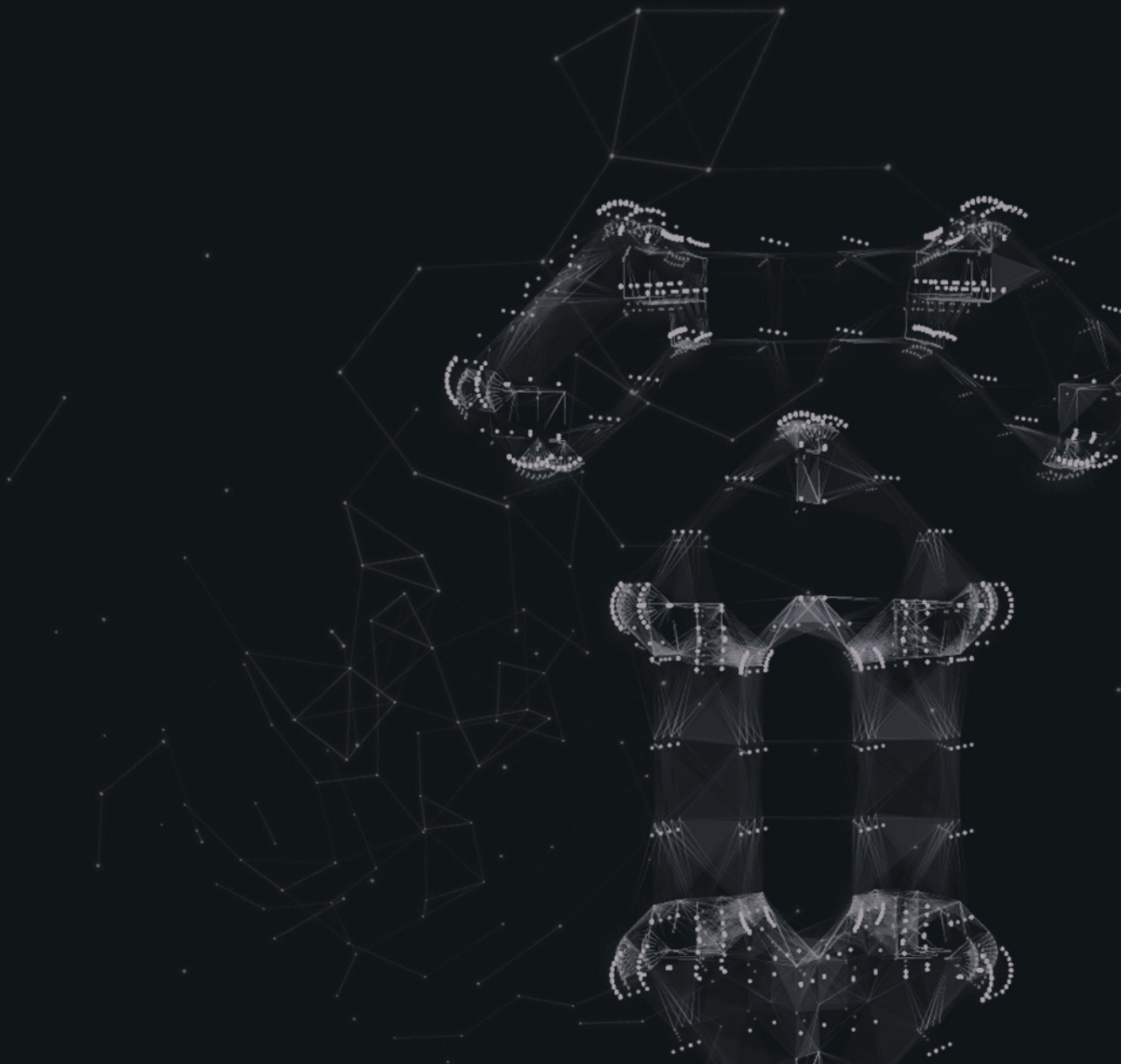
TENSET

Smart Contract Audit

# BURN CAT



# Table of Content

**Executive Summary** 3
**General security assumptions** 4
**Actors in the system** 5
**Thread model for each actor** 7
**Findings** 9
MMC-1 Pack variables into one slot for optimization Resolved 10
ERR-1 Incorrect NatSpec Documentation Resolved 11
**Disclaimer** 12

# Executive Summary

## Project Summary

| Language | Codebase | Commits |
| --- | --- | --- |
| Solidity | Private | 16278f4699a7a55d87efc6db6cb7a100a3b9f902 |

## Audit Scope

| ID | File | SHA256 |
| --- | --- | --- |
| MMC | Memecoin.sol | f7004995462acfbd0271c61c32270 653a6bf9e2f87bb1a88dbf6d0218c104f3c |
| MTB | MemecoinTaxBurner.sol | 940a9634539d0d2374fcad37d3a43 a77871638486d5b356116dbca75d5adbe28 |
| MTC | MemecoinTaxConsumer.sol | 6e4c1d3c8e591960ba046d2c2e079 807d1cd984b421766a8175411ed5992958c |
| IMC | IMemecoinTaxConsumer.sol | c32854cf35129600da5c713e3dc1d cdd82b25c323e6552a1e44a71ddc92419bd |
| ERR | Errors.sol | 88f54b98c7c6eb25102b5711a59b0 ae14488a6bf9b31d7b4198eef0b29232aa5 |
| IBR.sol | IERC20Burnable.sol | e3e9cf13e763f6b42e9c7554c8d89 199722a58f19539e25523ca47958174d8d9 |

# General security assumptions

In conducting our security audit of the BURN CAT token, we have made a number of important security assumptions. These assumptions are fundamental to our analysis and should be considered when evaluating the overall security and potential vulnerabilities of the system:

**Security of the Underlying Blockchain**
Our security analysis is predicated on the assumption that the underlying blockchain (for example, Ethereum) is secure and functions as intended. Potential vulnerabilities or flaws in the blockchain protocol are outside the scope of this audit.

**Trusted Environment**
We assume that users of the smart contract will interact with it in a secure and trusted environment. This encompasses the use of secure, up-to- date software and hardware, which is free from malware or any potential interference from malicious third parties.

**Private Key Security**
Users of the smart contract are assumed to safeguard their private keys effectively. This involves not sharing private keys, securely storing key backups, and using hardware wallets or other secure means for key management.

These security assumptions form the basis of our audit. Deviations from these assumptions could potentially lead to risks or vulnerabilities not covered in this analysis. Therefore, the effective management of these factors is critical to ensure the ongoing security of the BURN CAT token.

# Findings

## Audit Overview


## Issues

| Severity | Found | Resolved | Partially Fixed | Acknowledged |
|---|---|---|---|---|
| Critical | 0 | 0 | 0 | 0 |
| High | 0 | 0 | 0 | 0 |
| Medium | 0 | 0 | 0 | 0 |
| Low | 0 | 0 | 0 | 0 |
| Informational | 0 | 2 | 0 | 0 |
| **Total** | **0** | **2** | **0** | **0** |

## MMC-1
# Pack variables into one slot for optimization

Informational Resolved

### Description

Variables 'buybackStrategy and 'supervisedTransfersEndAt' can be optimized for storage efficiency. These variables can be packed into a single storage slot, reducing gas costs. Additionally, changing the data type of 'supervisedTransfersEndAt' to 'uint64' can further optimize storage usage, as it only requires 8 bytes compared to the 20 bytes required for an address.

### Recommendation

- Pack the variables 'buybackStrategy' and 'supervisedTransfersEndAt' into a single storage slot to reduce gas costs.

- Change the data type of 'supervisedTransfersEndAt' to 'uint64 to optimize storage usage.

## ERR-1
# Incorrect NatSpec Documentation

Informational Resolved

### Description

In the Errors.sol, the NatSpec documentation for the 'UnacceptableReference error states that the given reference is 'address(O)'. However, analysis of the Catamoto.sol contract at line 55 reveals that the reference may not always be 'address(O)'. Therefore, the NatSpec comment is inaccurate and should be corrected to provide a more appropriate description of the error condition.

### Recommendation

Update the NatSpec comment for the 'UnacceptableReference error in Errors.Sol to accurately reflect the error condition. Instead of stating that the given reference is address(0)', describe the error as 'Given reference is unsupported' to align with the actual behavior observed in the contract.

# Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted to and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, Consultant cannot guarantee the explicit security of the audited smart contracts.