TENSET

Smart Contract Audit
Catamoto token

Table of Content

Executive Summary

Project Summary

Language	Codebase	Commits
Solidity	Private	82fbb691728e519caa04c183a4c1beeaa69ea975

Binance Smart Chain
0x4b360c05860c3d0040eba90f00870fbfe9bc61b8

Audit Scope

ID	File	SHA256
CM	Catamoto.sol	755e1f00a9bde5d6526f3e94f4290 7292348e04a89794ac5bbf0158debb17523
ICT	ICatamotoTaxConsumer.sol	19a18e359471f2cce43768a443c07 64bf694c447d6bc3d94e85f6b97846363cd
CPB	CatamotoPancakeV2Buyback.sol	4ac2bcf49b328a4515006adc3f73e 253e92cd0669c490f57b21844454fe81694
CPL	CatamotoPancakeV2AutoLiquidity.sol	399078ad7af9d2cfb29253dae3d1a 172a072c2400049fa25bcb5be1b048445ba
CTC	CatamotoTaxConsumerPancakeV2.sol	a9ebfe19841f01f3158f184d46968 5b7ef05ca87eb120c30b1d86f9529085def
IPR	IPancakeRouterV2.sol	c81b6caad91461b0248e61a175ab7 d7023592b52a1a735244804248c29aca42c
IPF	IPancakeFactoryV2.sol	a4a1e2b819563698b51ea99d70d10 14c8dd0173db0f81bbe8d91ca052025dbe8
ERR	Errors.sol	c17947de0232eae362c43bb6e9b4b 7b516205b7dec6433fd3a3fccb014a177a3
WTH	Withdrawable.sol	7f8f67392f4c0e13ef31506b4b7d2 76c6217f3a2c35496603e788e250c4b919a

General security assumptions

In conducting our security audit of the Catamoto 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:

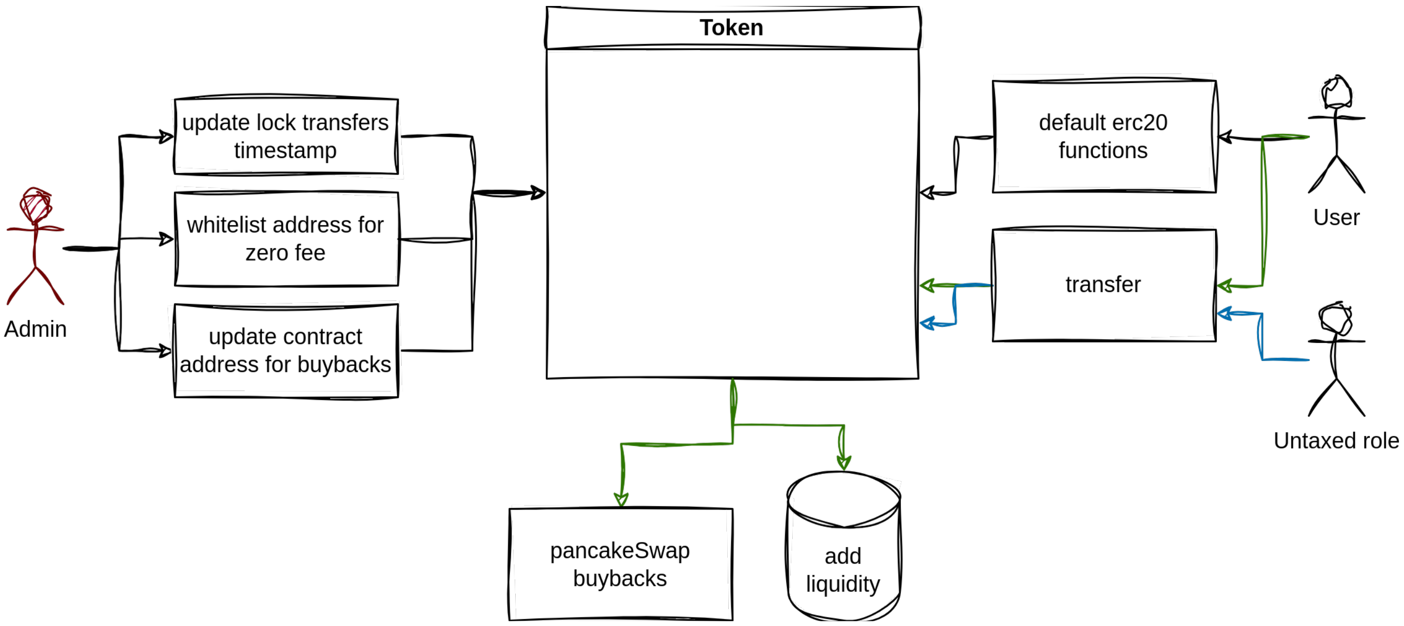
Security of the Underlying Blockchain
Our security analysis is predicated on the assumption that the underlying blockchain (for example, Ethereum) is secure and functions as intended. Potential vulnerabilities or flaws in the blockchain protocol are outside the scope of this audit.

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 Catamoto Token.

System Overview


Token
update lock transfers timestamp
whitelist address for zero fee
update contract address for buybacks
Admin
default erc20 functions
transfer
User
Untaxed role
pancakeSwap buybacks
add liquidity

Findings

Audit Overview

2
TOTAL ISSUES
Critical
High
Medium
Low
2 Informational

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**

CM-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(0)`. However, analysis of the Catamoto.sol contract at line 55 reveals that the reference may not always be `address(0)`. 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.