AegisX

Smart Contract Audits | KYC

GOLD

Security Assessment

FAT CAT Token

November 11, 2022

Table of Contents

Assessment Summary

This report has been prepared for FAT CAT Token on the BNB Chain network. AegisX provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- 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.

Project Overview

Contract Summary

Parameter	Result
Address	0x55493E35e33Fcf811571707Ac5Bf1DbcB658bAfc
Name	FAT CAT
Token Tracker	FAT CAT (FATCAT)
Decimals	9
Supply	1,000,000,000,000
Platform	BNB Chain
compiler	v0.8.4+commit.c7e474f2
Contract Name	LiquidityGeneratorToken
Optimization	200
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/address/0x55493E35e33Fcf811571707Ac5Bf1DbcB658bAfc#code
Payment Tx	

Project Overview

Risk Analysis Summary

Parameter	Result
Buy Tax	1.9%
Sale Tax	1.9%
Is honeypot?	Clean
Can edit tax?	Yes
Is anti whale?	No
Is blacklisted?	No
Is whitelisted?	Yes
Holders	Owner holds 25% Unlocked Tokens.
Security Score	65/100
Auditor Score	69/100
Confidence Level	Low

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Main Contract Assessed
Contract Name

Name	Contract	Live
FAT CAT	0x55493E35e33Fcf811571707Ac5Bf1DbcB658bAfc	Yes

TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
LiquidityGeneratorToken	da39a3ee5e6b4b0d3255bfef95601890afd80709	LiquidityGeneratorToken.sol

Mint Check

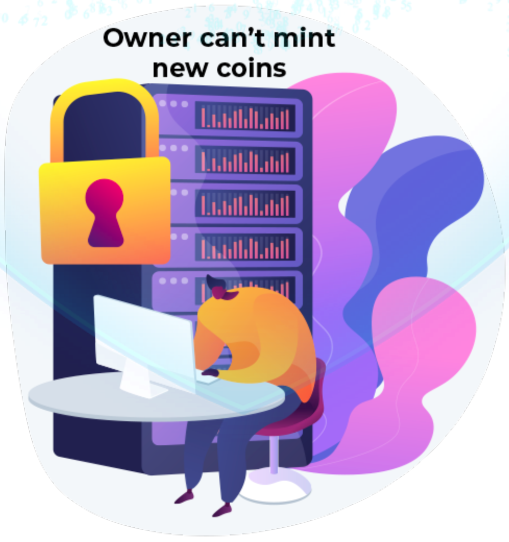
The project owners of FAT CAT do not have a mint function in the contract, owner cannot mint tokens after initial deploy.

The Project has a Total Supply of 1,000,000,000,000 and cannot mint any more than the Max Supply.

Mint Notes:

Auditor Notes: No Mint Function.

Project Owner Notes:



Fees Check

The project owners of FAT CAT do not have the ability to set fees higher than 25%.

The team May have fees defined; however, they can't set those fees higher than 25% or may not be able to configure the same.

Tax Fee Notes:

Auditor Notes: The contract currently charges a buy/sell fee of 1.9% and does have a function to change it but is capped at 25%.

Project Owner Notes: .

Blacklist Check

The project owners of FAT CAT do not have a blacklist function their contract.

The Project allow owners to transfer their tokens without any restrictions.

Token owner cannot blacklist the contract: Malicious or compromised owners can trap contracts relying on tokens with a blacklist.

Blacklist Notes:

Auditor Notes: The contract does not have a ban function. However it does have a function to exclude addresses from reflections.

Project Owner Notes:

MaxTx Check

The Project Owners of FAT CAT cannot set max tx amount

The Team allows any investors to swap, transfer or sell their total amount if needed.

MaxTX Notes:

Auditor Notes: Does not have a max tx limit function.

Project Owner Notes:

Pause Trade Check

The Project Owners of FAT CAT don't have the ability to stop or pause trading.

The Team has done a great job to avoid stop trading, and investors has the ability to trade at any given time without any problems

Pause Trade Notes:

Auditor Notes: Does not have a pause trade function.

Project Owner Notes:

Contract Ownership

The contract ownership of FAT CAT is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x992f8736641d15f33cb7b16bf949ad9e3239a978 which can be viewed:
HERE

The owner wallet has the power to call the functions displayed on the privileged functions chart below, if the owner's wallet is compromised, they could exploit these privileges.

We recommend the team renounce ownership at the right time, if possible, or gradually migrate to a timelock with governing functionalities regarding transparency and safety considerations.

We recommend the team use a Multisignature Wallet if the contract is not going to be renounced; this will give the team more control over the contract.

Liquidity Ownership

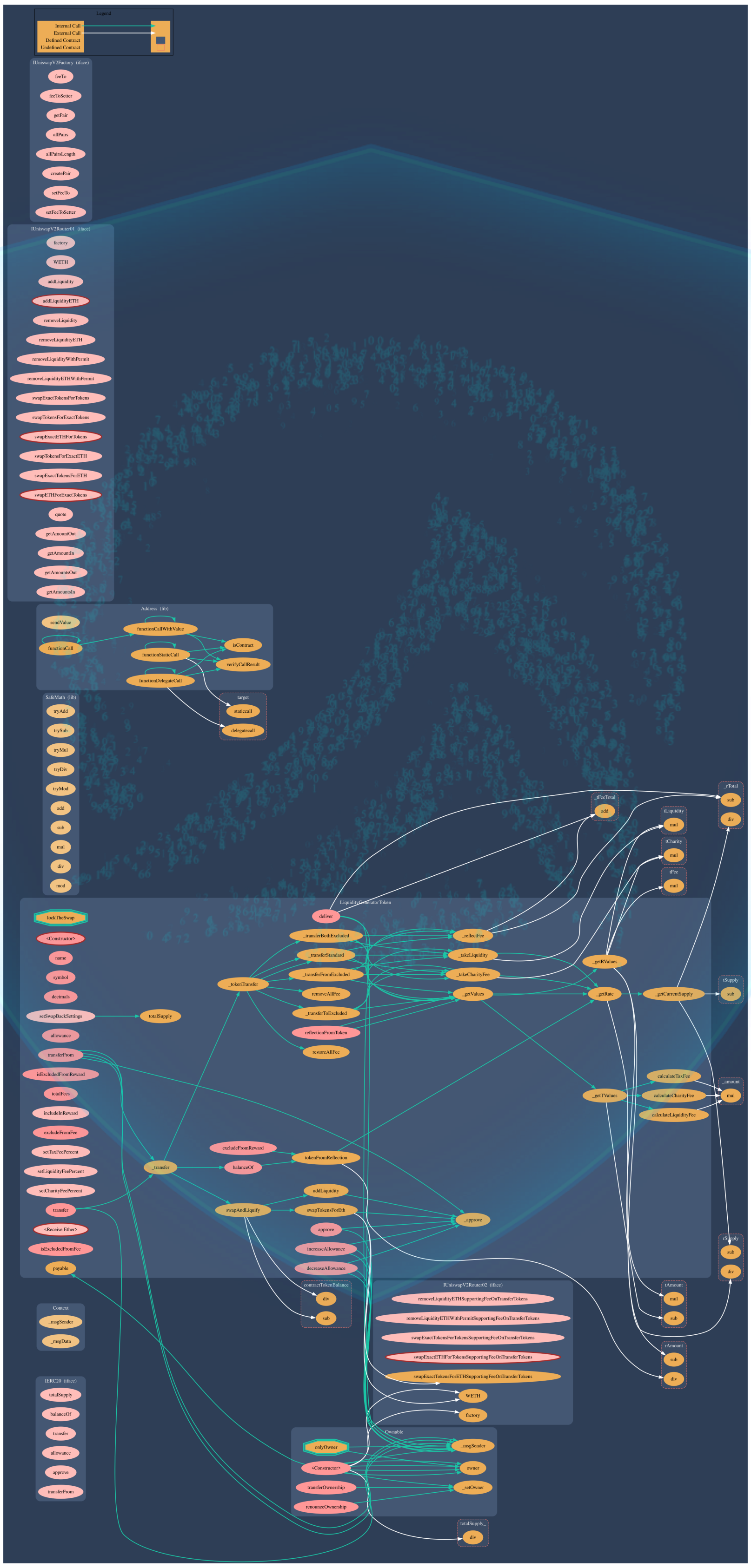
The token does not have liquidity at the moment of the audit, block 22935369

If liquidity is unlocked, then the token developers can do what is infamously known as 'rugpull'. Once investors start buying token from the exchange, the liquidity pool will accumulate more and more coins of established value (e.g., ETH or BNB or Tether). This is because investors are basically sending these tokens of value to the exchange, to get the new token. Developers can withdraw this liquidity from the exchange, cash in all the value and run off with it. Liquidity is locked by renouncing the ownership of liquidity pool (LP) tokens for a fixed time period, by sending them to a time-lock smart contract. Without ownership of LP tokens, developers cannot get liquidity pool funds back. This provides confidence to the investors that the token developers will not run away with the liquidity money. It is now a standard practice that all token developers follow, and this is what really differentiates a scam coin from a real one.

Read More

Call Graph

The contract for FAT CAT has the following call graph structure.



KYC Information

The Project Owners of FAT CAT are not KYC'd. .

The owner wallet has the power to call the functions displayed on the priviliged functions chart below, if the owner wallet is compromised this privileges could be exploited.

We recommend the team to renounce ownership at the right timing if possible, or gradually migrate to a timelock with governing functionalities in respect of transparency and safety considerations.

KYC Information Notes:

Auditor Notes: N/A

Project Owner Notes:

Smart Contract Vulnerability Checks

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	LiquidityGeneratorToken.sol	L: 959 C: 9
SWC-109	Pass	Uninitialized Storage Pointer.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	LiquidityGeneratorToken.sol	L: 0 C: 0

ID	Severity	Name	File	locatio n
SWC-112	Pass	Delegate Call to Untrusted Callee.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	LiquidityGeneratorT oken.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	LiquidityGeneratorT oken.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-126	Pass	Insufficient Gas Griefing.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	LiquidityGeneratorToken.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	LiquidityGeneratorToken.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-108 - State Variable Default Visibility

CWE-710: Improper Adherence to Coding Standards

Description:

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

Remediation:

Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

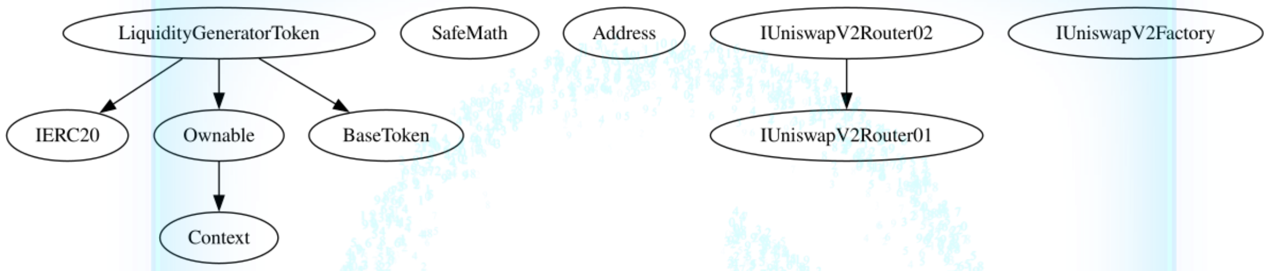
References:

Ethereum Smart Contract Best Practices - Explicitly mark visibility in functions and state variables

Inheritance

The contract for FAT CAT has the following inheritance structure.

The Project has a Total Supply of 1,000,000,000,000



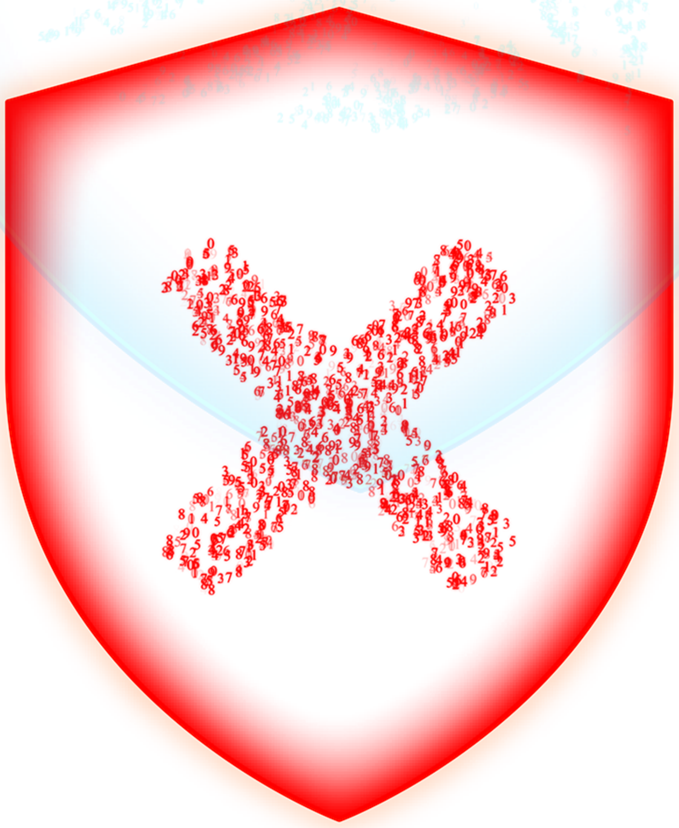
Privileged Functions (onlyOwner)

Function Name	Parameters	Visibility
renounceOwnership	none	Public
transferOwnership	none	Public
setAutoSwapBack	none	External
setNotify	none	External
setTradeStatus	none	External
setAutoLiquidityInterval	none	External
setAutoAddLiquidity	none	External
setDaoAddress	none	External
setTreasuryAddress	none	External
setFeeReceivers	none	External
setWhitelist	none	External
setBlacklist	none	External

Assessment Results

- The smart contract was generated using Pinksale.finance token generator.
- There is a buy/sell fee of 1.9% which can be changed up to 25%.
- The owner cannot ban an address but can exclude an address from reflections.
- The owner can whitelist an address an exempt from fees.
- The owner holds 25% of the supply unlocked.
- The owner's wallet was supplied via tornado cash to pay the fees for pinksale for the token generating and fundraising pool.

Audit Failed



Social Media Checks

Social Media	URL	Result
Website	http://fatcat.army/	Pass
Telegram	https://t.me/fatcatcoinchat	Pass
Twitter	https://twitter.com/fatcat_coin	Pass
OtherSocial	https://www.youtube.com/channel/UC22v-zVmDZidbza422fImpg	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

AegisX has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and AegisX is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will AegisX or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by AegisX are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.