



AegisX

Smart Contract Audits | KYC



Audit Status: Pass

PALLADIUM

Security Assessment

FomoVenom Token

June 12, 2023

Table of Contents

1 Assessment Summary

2 Technical Findings Summary

3 Project Overview

3.1 Token Summary

3.2 Risk Analysis Summary

3.3 Main Contract Assessed

4 Smart Contract Risk Checks

4.1 Mint Check

4.2 Fees Check

4.3 Blacklist Check

4.4 MaxTx Check

4.5 Pause Trade Check

4.6 Contract Ownership

4.7 Liquidity Ownership

4.8 KYC Check

5 Smart Contract Vulnerability Checks

5.1 Smart Contract Vulnerability Details

5.2 Smart Contract Inheritance Details

5.3 Smart Contract Privileged Functions

6 Assessment Results and Notes(Important)

7 Social Media Checks(Informational)

8 Technical Findings Details

9 Disclaimer

Assessment Summary

This report has been prepared for FomoVenom 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.

Technical Findings Summary

Classification of Risk

Severity	Description
● Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
● Major	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
● Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
● Minor	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
● Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
● Critical	0	0	0
● Major	0	0	0
● Medium	0	0	0
● Minor	2	0	2
● Informational	2	0	2
Total	4	0	4

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
● Critical	Danger or Potential Problems.
● Major	Be Careful or Fail test.
● Minor	Pass, Not-Detected or Safe Item.
● Informational	Function Detected

Manual Code Review Risk Results

Contract Priviledge	Description
● Buy Tax	3% (25% Max)
● Sale Tax	3% (25% Max)
● Cannot Sale	Pass
● Cannot Sale	Pass
● Max Tax	25
● Modify Tax	Up to 25%
● Fee Check	Pass
● Is Honeypot?	Not Detected
● Trading Cooldown	Not Detected
● Can Pause Trade?	Pass
● Pause Transfer?	Not Detected
● Max Tx?	Pass

Contract Priviledge	Description
ⓘ Is Anti Whale?	Not Detected
ⓘ Is Anti Bot?	AntiSniper and AntiMEV detected
ⓘ Is Blacklist?	Blacklist conjunction with AntiBot detected
ⓘ Blacklist Check	Pass
ⓘ is Whitelist?	Detected
ⓘ Can Mint?	Pass
ⓘ Is Proxy?	Not Detected
ⓘ Can Take Ownership?	Not Detected
ⓘ Hidden Owner?	Not Detected
ⓘ Owner	
ⓘ Self Destruct?	Not Detected
ⓘ Other?	Not Detected
ⓘ Other?	Not Detected
ⓘ Holders	1
ⓘ Auditor Confidence	High

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.

Project Overview

Contract Summary

Parameter	Result
Address	
Name	FomoVenom
Token Tracker	FomoVenom (FVNM)
Decimals	18
Supply	470,000,000,000,000
Platform	BNB Chain
compiler	v0.8.19+commit.7dd6d404
Contract Name	FomoVenom
Optimization	999,999,999
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	

Project Overview

Simulation Summary

Parameter	Result
Transfer From Owner	Pass
Transfer From Holder	Pass
Add Liquidity	Pass
RemoveLiquidity	Pass
Buy from Owner	Pass
Buy from Holder	Pass
Sale from Owner	Pass
Sale from Holder	Pass
Remove Liquidity	Pass
SwapAndLiquify	Pass
SwapAndSale w/Fee	Pass
SwapAndSale TX	
SwapAndSaleNoFee	Pass
SwapAndSale No/Fee TX	
ExcludeFromFees	Pass
LaunchPad	PinkSale

Parameter	Result
Pool Creation	Pass
Pool Creation TX	
Pool Finalize	Pass
Pool Finalize TX	
Enable	Pass

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.

Project Overview

Advance Verification

Parameter	Result
Transfer From Owner	Pass
Transfer From Holder	Pass
Add Liquidity	Pass
RemoveLiquidity	Pass
Buy from Owner	Pass
Buy from Holder	Pass
Sale from Owner	Pass
Sale from Holder	Pass
SwapAndLiquify	Pass
LaunchPad	PinkSale

The following is a simulation of the contract in our local testnet or using one of the public testnet environments, this is to ensure the contract operations are fully functional and that are able to pass the specific launchpad criterias. While this section may be a pass, please understand you need to review all other details of the project. Always DYOR.

MainNet Contract was Not Assessed

TestNet Contract Assessed Contract Name

Name	Contract	Live
FomoVenom	0xa7FDOAEdf6CC4D551076E0af60D481Fd780c001F	Yes

Solidity Code Provided

SolidID	File Sha-1	FileName
FomoVenom	b551da639d50c370e57710e4605a147401ee0c72	FomoVenom.sol

Mint Check

The project owners of FomoVenom do not have a mint function in the contract, owners cannot mint tokens after initial deployment.

The Project has a Total Supply of 470,000,000,000,000 and cannot mint any more than the Max Supply.

Mint Notes:

Auditor Notes: A mint function only gets called in the constructor.

Project Owner Notes:



Fees Check

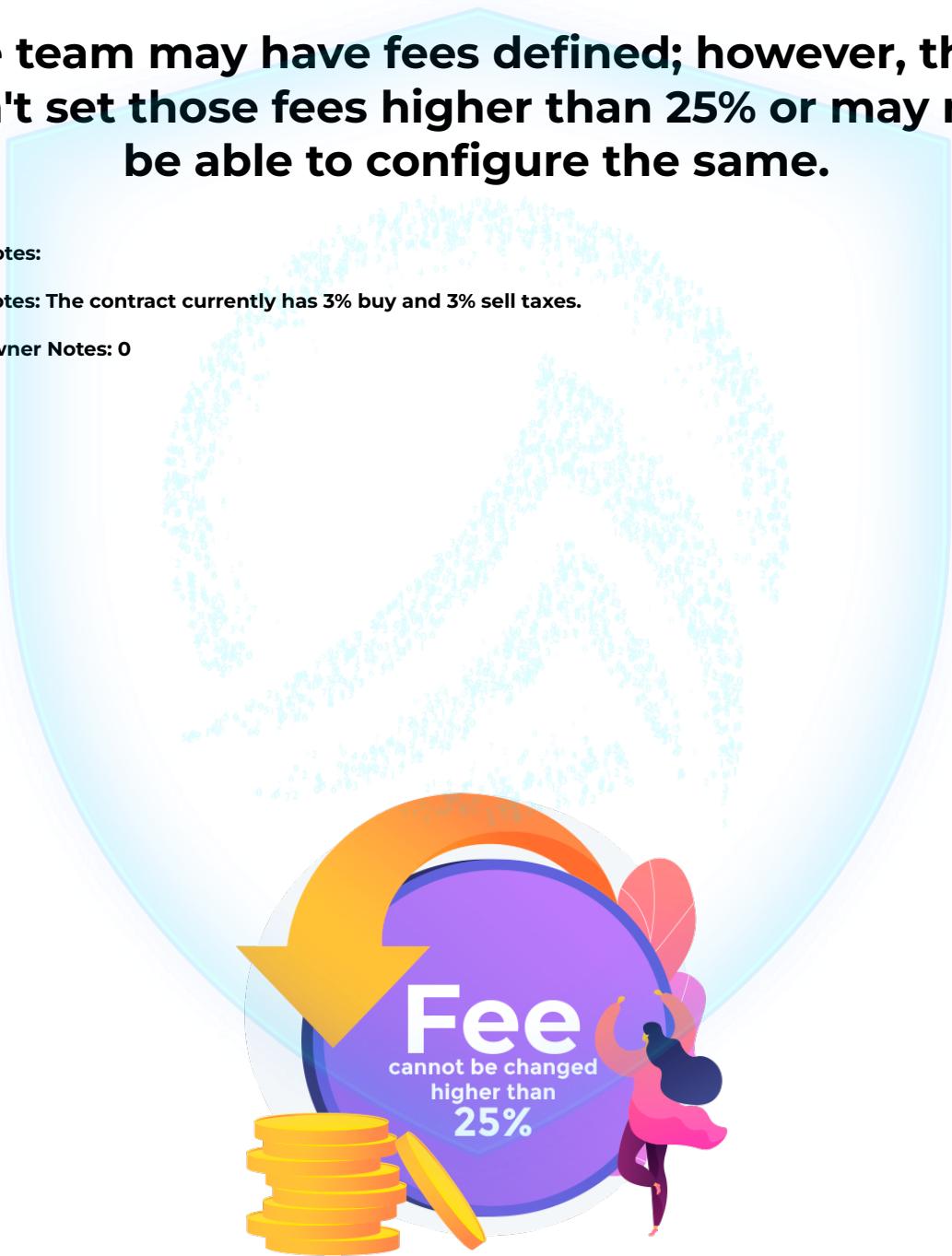
The project owners of FomoVenom do not have the ability to set fees higher than 25%.

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 has 3% buy and 3% sell taxes.

Project Owner Notes: 0



Blacklist Check

The project owners of FomoVenom do not have a blacklist function on 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 have a blacklist function in conjunction with AntiBot.

Project Owner Notes:



MaxTx Check

The Project Owners of FomoVenom cannot set max tx amount

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

MaxTX Notes:

Auditor Notes: The contract does not have a max tx function.

Project Owner Notes:



Project has no MaxTX



Pause Trade Check

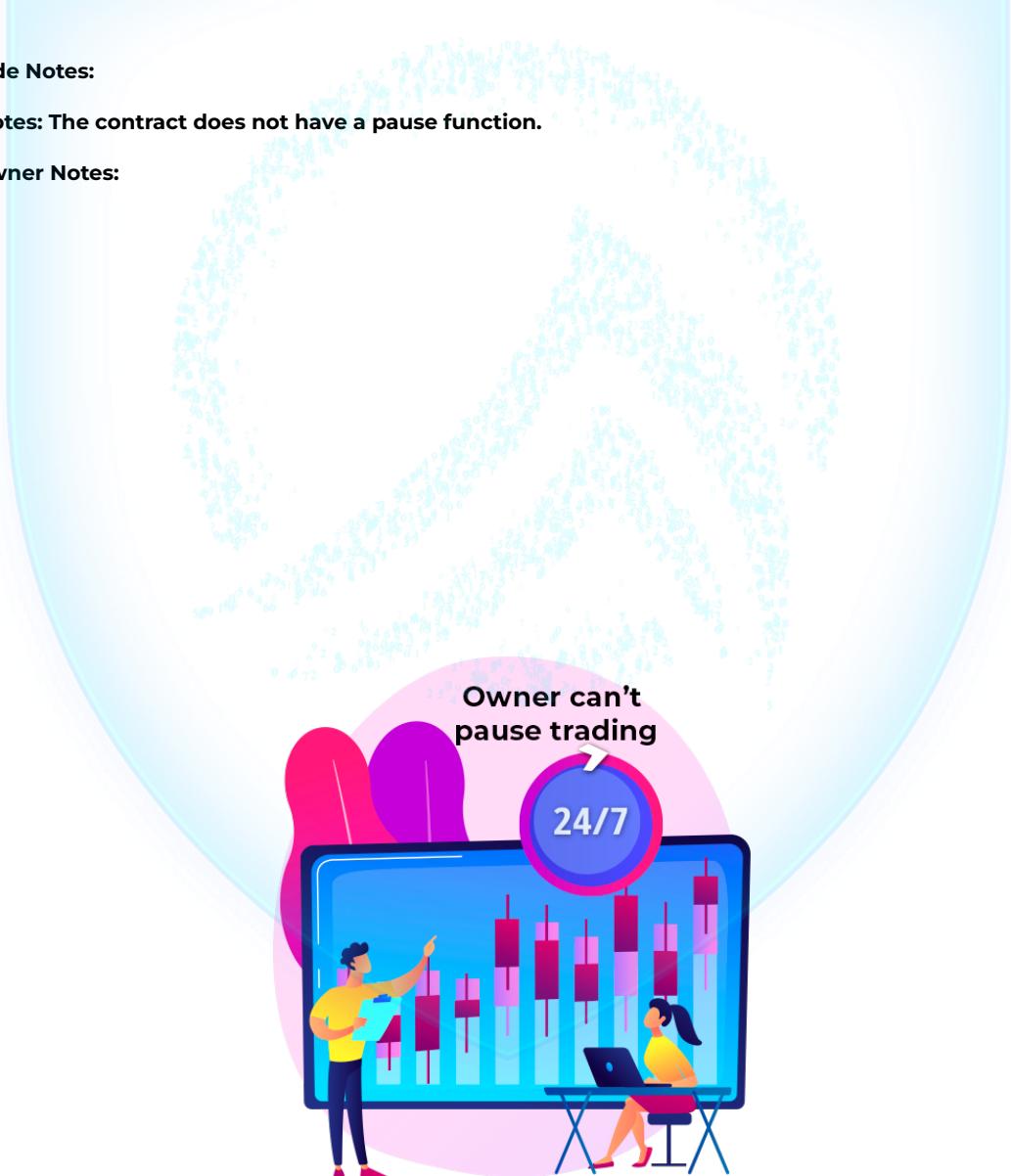
The Project Owners of FomoVenom don't have the ability to stop or pause trading.

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: The contract does not have a pause function.

Project Owner Notes:



Contract Ownership

The contract ownership of FomoVenom 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 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

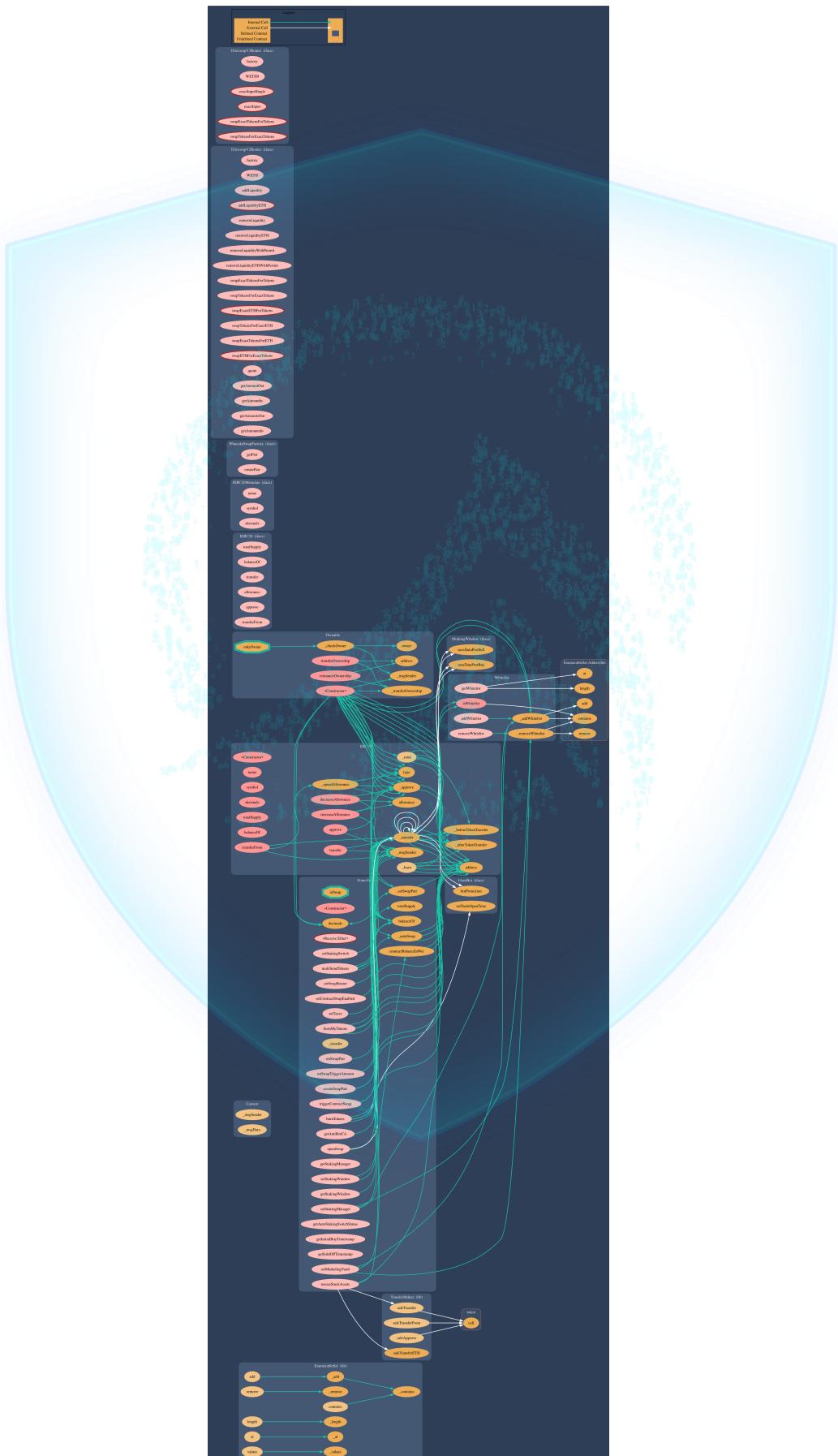
If liquidity is unlocked, then the token developers can do what is infamously known as 'rugpull'. Once investors start buying token from the exchange, the liquidity pool will accumulate more and more coins of established value (e.g., ETH or BNB or Tether). This is because investors are basically sending 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 FomoVenom has the following call graph structure.



KYC Information

The Project Owners of FomoVenom are not KYC'd. .

The owner wallet has the power to call the functions displayed on the privileged 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:

Project Owner Notes:



Smart Contract Vulnerability Checks

ID	Severity	Name	File	Location
SWC-100	Pass	Function Default Visibility	FomoVenom.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	FomoVenom.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	FomoVenom.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	FomoVenom.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	FomoVenom.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	FomoVenom.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	FomoVenom.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	FomoVenom.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	FomoVenom.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	FomoVenom.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	FomoVenom.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	FomoVenom.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	FomoVenom.sol	L: 0 C: 0

ID	Severity	Name	File	Location
SWC-113	Pass	Multiple calls are executed in the same transaction.	FomoVenom.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	FomoVenom.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	FomoVenom.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	FomoVenom.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	FomoVenom.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	FomoVenom.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	FomoVenom.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randomness.	FomoVenom.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	FomoVenom.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	FomoVenom.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	FomoVenom.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	FomoVenom.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	FomoVenom.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	FomoVenom.sol	L: 0 C: 0

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

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

Inheritance

The contract for FomoVenom has the following inheritance structure.

The Project has a Total Supply of 470,000,000,000,000

Privileged Functions (onlyOwner)

Function Name	Parameters	Visibility
renounceOwnership		Public
transferOwnership	address newOwner	Public
addWhitelist	address account	External
removeWhitelist	address account	External
getWhitelist	uint256 start, uint256 limit	External
rescueStuckAssets	bool ethOrToken, address tokenCA, uint256 amt, address receivable	External
multiSendTokens	address[] memory accounts, uint256[] memory amountsInWei	External
openSwap	address _antiBot	External
getAntiBotCA		External

Function Name	Parameters	Visibility
setSwapPair	address pairedCoin, address swapPair, bool enable	External
createSwapPair	address pairedCoin	External
setSwapRouter	address router, bool enable	External
setContractSwapEnabled	bool _enabled	External
setSwapTriggerAmount	uint256 newAmount	External
setTaxes	uint16 _marketingTax, uint16 _stakingTax	External
setMarketingVault	address _marketing	External
setStakingManager	address _ca	External
setStakingWindow	address _stakingWindow	External
setStakingSwitch	bool _switch	External

Function Name	Parameters	Visibility
burnTokens	uint256 amount	External



FVNM-01 | Potential Sandwich Attacks.

Category	Severity	Location	Status
Security	● Minor	FomoVenom.sol: 1569, 13	✓ Resolved

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by back running (after the transaction being attacked) a transaction to sell the asset. The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

Function Name	Slippage
_autoSwap	100

Remediation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

Project Action

Without an use of oracle, there is always a risk of potential sandwich attacks.

It was informed that there is an antibot measure in place, protecting against the sandwich attacks.

References:

[What Are Sandwich Attacks in DeFi – and How Can You Avoid Them?.](#)

FVNM-10 | Initial Token Distribution.

Category	Severity	Location	Status
Centralization / Privilege	 Informational	FomoVenom.sol: 1339,9	 Resolved

Description

All of the FomoVenom tokens are sent to the contract deployer when deploying the contract.

This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

Remediation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Project Action

The total supply gets minted at the time of the deployment and sent to the deployer.

It was explained that sending to a single address is unavoidable due to the nature of how the tokens launch. The risk was minimized by sending it to a specified kingVenom address instead of the owner or the deployer address.

FVNM-12 | Centralization Risks In The onlyOwner Role(s)

Category	Severity	Location	Status
Centralization / Privilege	● Minor	FomoVenom.sol: 990,5	● Resolved

Description

In the contract FomoVenom, the role `onlyOwner` has authority over the functions that lead to centralization risks.

Any compromise to the `onlyOwner` account(s) may allow the hacker to take advantage of this authority.

Remediation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage.

We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked.

In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets.

Project Action

`_owner` has a right to whitelist the addresses.

It was informed that the whitelisting capability had to be placed for its intended use for other dapps that requires whitelisting involved addresses.

FVNM-13 | Extra Gas Cost For User

Category	Severity	Location	Status
Logical Issue	i Informational	FomoVenom.sol: 1469,13	✓ Resolved

Description

The user may trigger a tax distribution during the transfer process, which will cost a lot of gas and it is unfair to let a single user bear it.

Remediation

We advise the client to make the owner responsible for the gas costs of the tax distribution.

Project Action

_autoSwap gets triggered when the threshold is met.

It was informed that the dev did the due diligence to minimize the gas burden on the investors by minimizing the codes used and optimizing the runs to 999,999,999.

Social Media Checks

Social Media	URL	Result
Website	https://www.fomovenom.com/	Pass
Telegram	https://t.me/fomovenom	Pass
Twitter	https://twitter.com/venomfomo3d	Pass
OtherSocial	https://GitHub.com/fomovenom/	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:

Assessment Results

Score Results

Review	Score
Overall Score	98/100
Auditor Score	89/100
Review by Section	Score
Manual Scan Score	16/18
SWC Scan Score	37/37
Advance Check Score	45/45

The maximum score is 100, however to attain that value the project must pass the reviews and provide all the data needed for the assessment. Minimum score to pass is 80 points. If a project fails to attain 80 and/or has unresolved critical and/or major and/or medium finding(s) in the Palladium tier assessments, an automatic failure is given. Read our notes and final assessment below.



PASSED

Assessment Results

Auditor Score = 89
Audit Passed



PASSED

Important Notes from the Auditor:

- Without an use of oracle, there is always a risk of potential sandwich attacks - It was informed that there is an antibot measure in place, protecting against the sandwich attacks.
-
- The total supply gets minted at the time of the deployment and sent to the deployer - It was explained that sending to a single address is unavoidable due to the nature of how the tokens launch. The risk was minimized by sending it to a specified kingVenom address instead of the owner or the deployer address.
-
- _owner has a right to whitelist the addresses - It was informed that the whitelisting capability had to be placed for its intended use for other dapps that requires whitelisting involved addresses.
-
- _autoSwap gets triggered when the threshold is met - It was informed that the dev did the due diligence to

minimize the gas burden on the investors by minimizing the codes used and optimizing the runs to 999,999,999.



Appendix

Finding Categories

Centralization / Privilege

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

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM 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 how `block.timestamp` works.

Control Flow

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

Volatile Code

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

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the 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 setter function.

Coding Best Practices

ERC 20 Coding Standards are a set of rules that each developer should follow to ensure the code meets a set of criteria 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 advocacy 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 depreciation 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.

