CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

April 17, 2022

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Scope of Work

ViceVersa Protocol team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **ViceVersa Protocol.**

Network

Binance Smart Chain (BEP20)

Contract link

https://bscscan.com/token/0x914a890423b487df9d54b948a5d26224267d6a92

Website

http://Viceversa.finance

Telegram

https://t.me/ViceVersaProtocol

Twitter

https://twitter.com/ViVaProtocol

Description

ViceVersa Protocol is a life changing DeFi 3.0 auto-staking and auto-rebasing Protocol. It’s a rebased auto-staking DeFi with multiply levels and play and earn features.

Logo

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.
Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that 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 ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
    - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

    - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Used Code from other Frameworks/Smart Contracts (Direct Imports)

Imported Packages

- SafeMathInt
- IERC20
- SafeMath
- InterfaceLP
- Roles
- MinterRole
- ERC20Detailed
- IDEXRouter
- IDEXFactory
- Ownable
- VIVA

Description

Optimization enabled: Yes

Decimal: 18

Symbol: VIVA

Max / Total supply: 4,000,000,000

Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	3	3	4	1

Exposed Functions

Version	Public	Private	External	Internal
1.0	18	1	40	24

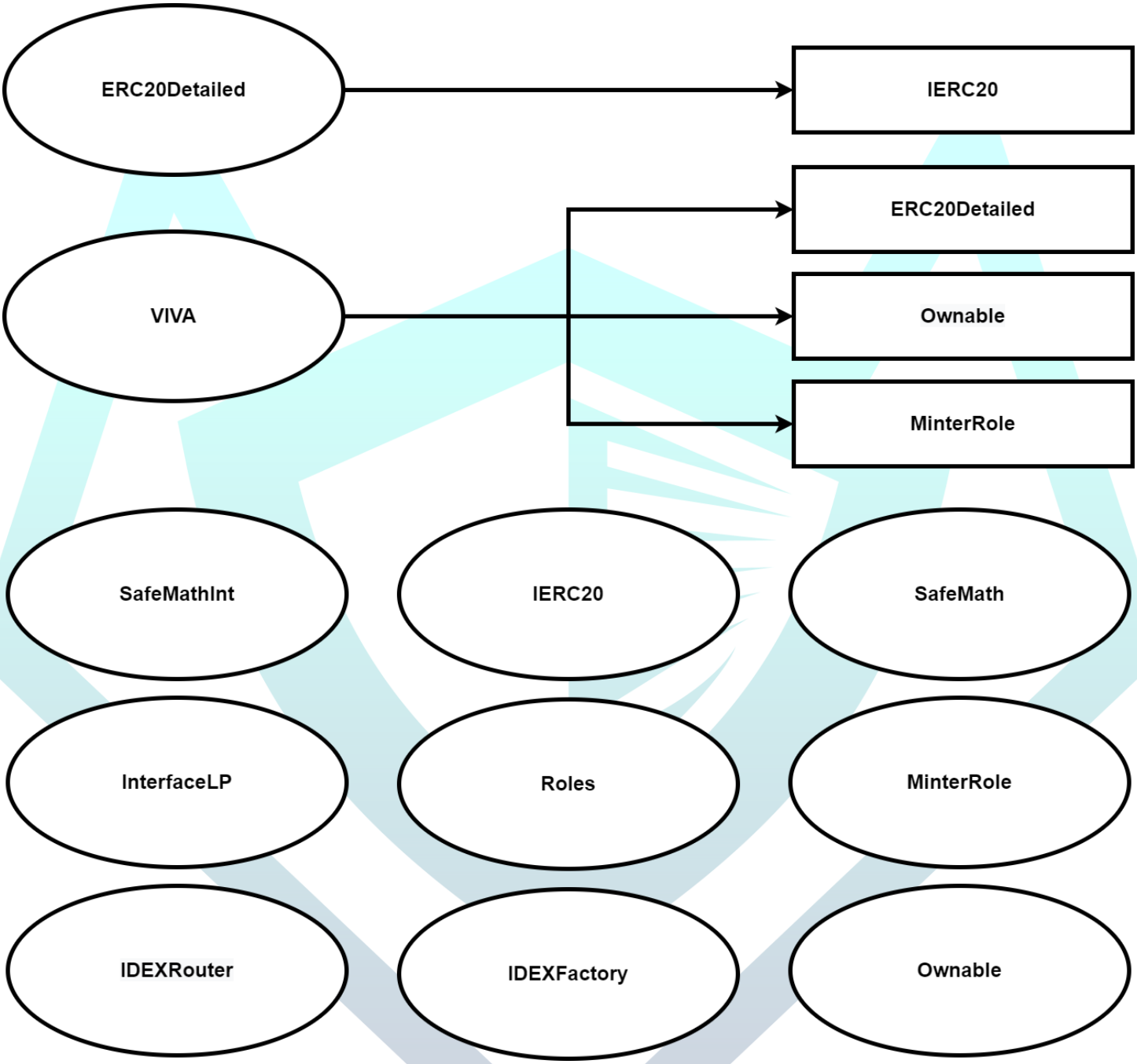
State Variables

Version	Total	Public
1.0	37	13

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.7.6		Yes	No	No

Inheritance Graph



Correct implementation of Token Standard

Tested	Verified
✓	✗

Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✓	✓	✓
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Verify Claims

Statement	Exist	Tested	Deployer
Renounce Ownership	✓	✓	✓
Mint	✓	✓	✗
Burn	–	–	–
Block	✓	✓	✓
Pause	–	–	–

Legend

Attribute	Symbol
Verified / Can	✓
Verified / Cannot	✗
Unverified / Not checked	⚑
Not Available	–

Write Functions of Contract

1. addMinter
2. approve
3. clearStuckBalance
4. decreaseAllowance
5. enableTransfer
6. increaseAllowance
7. manualSync
8. mint
9. rebase
10. removeMinter
11. renounceMinter
12. renounceOwnership
13. rescueToken
14. sendPresale
15. setFeeExempt
16. setFeeReceivers
17. setFees
18. setInitialDistributionFinished
19. setLP
20. setSwapBackSettings
21. setTargetLiquidity
22. transfer
23. transferFrom
24. transferOwnership
25. updateBlacklist

AUDIT PASSED

Low Issues

A floating pragma is set (SWC-103)	L: 40 C: 10
State variable visibility is not set (SWC-108)	L: 382 C: 29, L: 383 C: 29, L: 412 C: 12, L: 413 C: 12, L: 419 C: 12, L: 420 C: 12, L: 427 C: 9

Audit Comments

- Contract has rebase that can increase total supply
- Minter can mint tokens after initial deployment
- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can block user address
- Deployer can set rebase
- Deployer can set LP address
- Deployer can set initial distribution status as finished
- Deployer can enable transfer with an address
- Deployer can exclude address from fees
- Deployer can set swap back setting as enabled
- Deployer can set target liquidity
- Deployer can add/remove minter address
- Deployer can send tokens to multiple address
- Deployer can set fee receiver address
- Deployer can set fees not greater than 25%
- Deployer can collect tokens from contract
- Deployer cannot burn tokens
- Deployer cannot mint after initial deployment
- Deployer cannot pause contract

CONTRACTWOLF

Blockchain Security - Smart Contract Audits