CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

April 17, 2022

Disclaimer 3

Scope of Work & Engagement 3

Links 4

Project Description 5

Logo 5

Risk Level Classification 6

Methodology 7

Used Code from other Frameworks / Smart Contracts (Imports) 8

Token Description 12

Inheritance Graph 18

Overall Checkup 20

Verify Claim 21

Audit Result 22

Audit Comments 23

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 it's **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

Orcus Finance 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 **Orcus Finance.**

Network

Astar Network

Contract link

TBA (Local Files)

Website

https://orcusfinance.io

Discord

https://discord.com/invite/mQmYqHQ76E

Twitter

https://twitter.com/Orcus_Finance

Medium

https://orcusfinance.medium.com

Description

Orcus Finance (the "Protocol") is delivering the first Fractional-Algorithmic Stablecoin pegged to the United States dollar which is built on the Astar Network. Since Astar Network is a gateway to the multi-chain environment, we are going to expand our stablecoin to as many networks as possible in the near future.

Orcus Finance is an entirely decentralized and autonomous protocol with the native governance token which aims to be the first leading Fractional-Algorithmic Stablecoin issuer on the Astar network and to implement multiple useful financial tools and other synthetic assets in the ecosystem.

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

ERC20

- OrcusERC20
- ERC20
- OnlyBank
- IOrcusERC20
- ORU
- oUSD

_mock

- MockBankSafe
- MockERC20
- ERC20
- Ownable
- OrcusERC20
- MockOrcusERC20
- MockSwapController
- MockUSDC

aave

- DataTypes
- IFlashLoanReceiver
- ILendingPool
- ILendingPoolAddressProvider

Bank

- Arbitrager
- Bank
- BankRecollatStates
- BankSafe
- BankStates
- Dustbin
- FarmLPLock

- OrcusV1Distributor
- ProfitController
- Treasury
- OnlyBank
- OrcusProtocol
- EnableSwap
- IBankSafe
- Initializable
- IProfitController
- Farmable

Interfaces

- IBank
- IBankSafe
- IDIAOracleV2
- IFarm
- IFirebirdFactory
- IFirebirdFormula
- IFirebirdRouter
- IFirebirdZap
- IORUStake
- IOUSCERC20
- IPriceOracle
- IProfitController
- ISwapController
- ITwapOracle
- IUniswapV2Router

Libraries

- Babylonian
- FixedPoint
- TransferHelper
- UQ112x112

Oracles

- Ownable
- IDIAOracleV2
- IPriceOracle
- OrcusProtocol
- TwapOracle
- PriceOracle
- Console
- IUniswapV2Pair
- FixedPoint
- ITwapOracle
- TwapOracle

Sale

- Ownable
- IERC20
- SafeERC20
- IOUSDERC20
- Console
- ORUSale

Stake

- IERC20
- ERC20
- SafeERC20
- SafeMath
- Initializable
- IProfitController
- IORUStake
- OrcusProtocol
- OruStake

common

- EnableSwap
- Farmable
- OnlyArbitrager
- OnlyBank
- OrcusProtocol
- SwapController

Description

Optimization enabled: Yes

Capabilities

Components

ERC20				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	0	0	1

_mock				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	5	0	0	0

aave				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	0	1	4	0

Bank				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	10	0	0	0

Common				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	3	2	1	3

Farm				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	0	0	0

Interfaces				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	0	0	15	0

Libraries				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	0	4	0	0

Oracles				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	0	0	0

Sale				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	0	0	0

Stake				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	0	0	0

Exposed Functions

ERC20				
Version	Public	Private	External	Internal
1.0	6	0	6	1

_mock				
Version	Public	Private	External	Internal
1.0	12	0	4	0

aave				
Version	Public	Private	External	Internal
1.0	0	0	49	0

Bank				
Version	Public	Private	External	Internal
1.0	41	0	10	39

Common				
Version	Public	Private	External	Internal
1.0	13	6	13	9

Farm				
Version	Public	Private	External	Internal
1.0	19	2	3	2

Interfaces				
Version	Public	Private	External	Internal
1.0	0	0	91	0

Libraries				
Version	Public	Private	External	Internal
1.0	0	0	13	3

Oracles				
Version	Public	Private	External	Internal
1.0	4	0	4	1

Sale				
Version	Public	Private	External	Internal
1.0	5	0	0	0

Stake				
Version	Public	Private	External	Internal
1.0	7	0	4	17

State Variables

ERC20		
Version	Total	Public
1.0	16	14

_mock		
Version	Total	Public
1.0	9	9

aave		
Version	Total	Public
1.0	0	0

Bank		
Version	Total	Public
1.0	80	70

Common		
Version	Total	Public
1.0	32	20

Farm		
Version	Total	Public
1.0	17	13

Interfaces		
Version	Total	Public
1.0	0	0

Libraries		
Version	Total	Public
1.0	4	0

Oracles		
Version	Total	Public
1.0	18	10

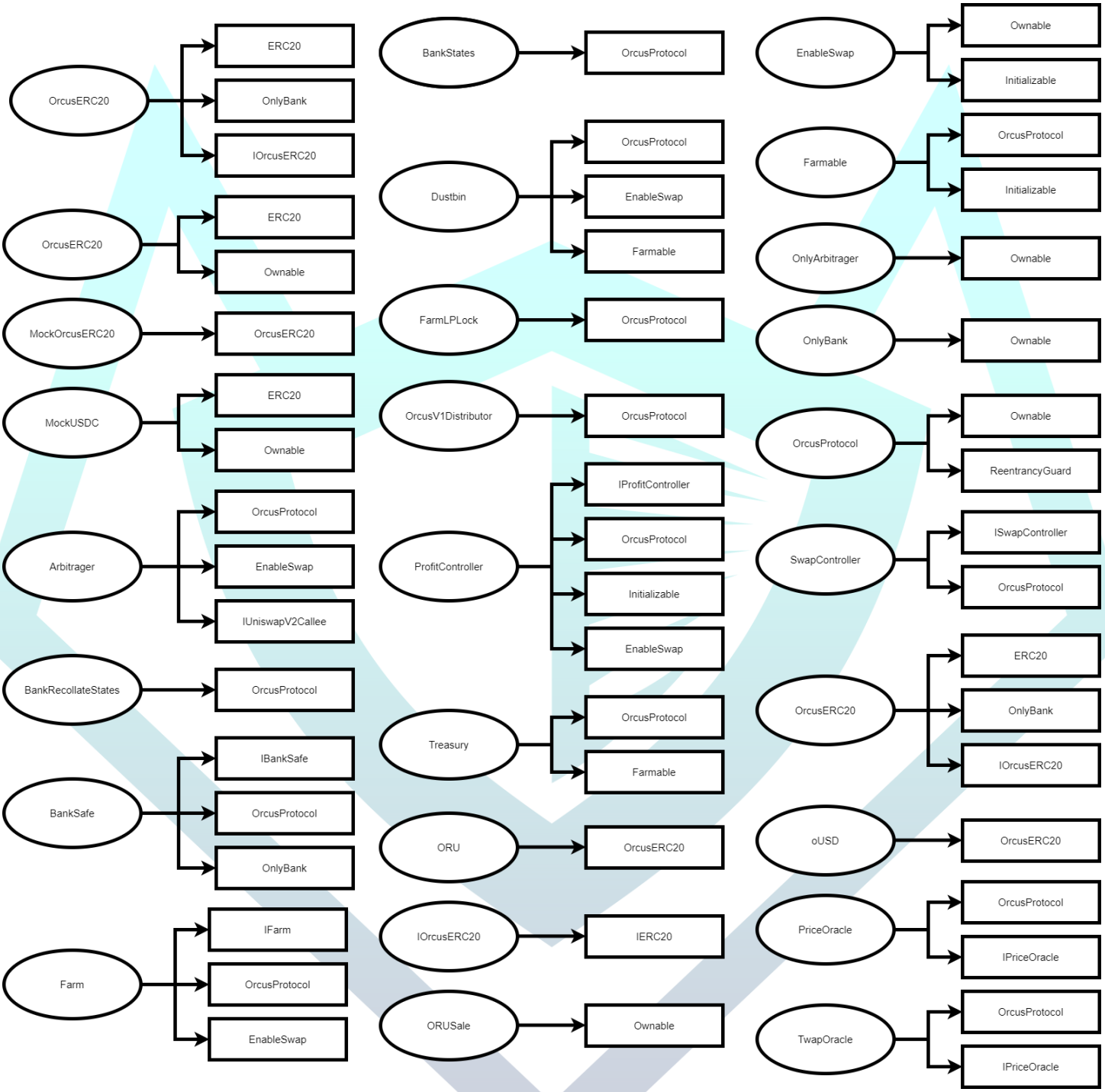
Sale		
Version	Total	Public
1.0	0	9

Stake		
Version	Total	Public
1.0	0	7

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.4		Yes	No	No

Inheritance Graph



MockSwapController
IWETH
IFirebirdRouter
MockBankSafe
Babylonian
IFirebirdZap
DataTypes
SafeMath
IUniswapV2Router
IFlasLoanReceiver
IBank
TransferHelper
IAaveIncentivesController
IBankSafe
UQ112x112
ILendingPool
IDIAOracleV2
IFirebirdFactory
ILendingPoolAddressesProvider
IFirebirdFormula
Bank

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	–

AUDIT PASSED

Audit Comments

ERC20

- Deployer can add/remove burn addresses
- Deployer can add/remove farm addresses
- Deployer can set treasury address
- Burn addresses can burn tokens
- Bank addresses can mint tokens after initial deployment
- Farm addresses can mint tokens after initial deployment

Bank

- Deployer can set contract addresses
- Deployer or operator can set target band
- Deployer or operator can set swap fee with an indefinite amount
- Deployer or operator can buy / sell OUSD
- Deployer or operator can set recollat
- Deployer or operator can set pool ceiling
- Deployer or operator can set rct per hour
- Deployer or operator can collect profit from contract
- Deployer or operator can burn liquidity pool
- Deployer or operator can set oru and oru pair address
- Deployer or operator can harvest farm
- Deployer or operator can claim farm
- Deployer or operator can burn liquidity pool
- Deployer or operator can set oru and oru pair address
- Deployer or operator can harvest farm
- Deployer or operator can claim farm
- Deployer can set profit controller
- Deployer can set idle collateral utilization ratio
- Deployer can set reserved collateral threshold
- Deployer can set excess collateral safety margin
- Deployer can set aave lending pool

- Deployer can set mint fee with an indefinite amount
- Deployer can set redeem fee with an indefinite amount
- Deployer can set swap fee with an indefinite amount
- Deployer can set swap fee with an indefinite amount
- Deployer can deposit farm and lock liquidity pool
- Deployer can withdraw and harvest farm
- Deployer can set farm addresses
- Deployer can set claimer addresses
- Deployer can execute transactions
- Deployer can convert profits to tokens
- Deployer can distribute stake
- Deployer can set oru stake
- Deployer can set burn rate
- Deployer can collect tokens from contract

Common

- Deployer or operator can farm deposit
- Deployer or operator can farm harvest
- Deployer or operator can farm claim
- Deployer can set farm address
- Deployer can transfer tokens from contract
- Deployer can set arbitrager address
- Deployer can set bank address
- Deployer can set operator address
- Deployer can set contract addresses
- Deployer can set pair paths
- Deployer can set dex IDss

Farm

- Deployer can add liquidity pool
- Deployer can set allocation in pool
- Deployer can set oru per second

- Deployer can set vesting penalty
- Deployer can set oru pair address
- Deployer can set bank safe address
- Deployer can set treasury address

Oracles

- Deployer can set OUSD oracle
- Deployer can set oru oracle
- Deployer can set period time

Sale

- Deployer can set start sale
- Deployer can set finish sale
- Deployer can collect tokens from contract
- Deployer can set profit controller
- Deployer can enable stake pause
- Deployer can collect tokens

CONTRACTWOLF

Blockchain Security - Smart Contract Audits