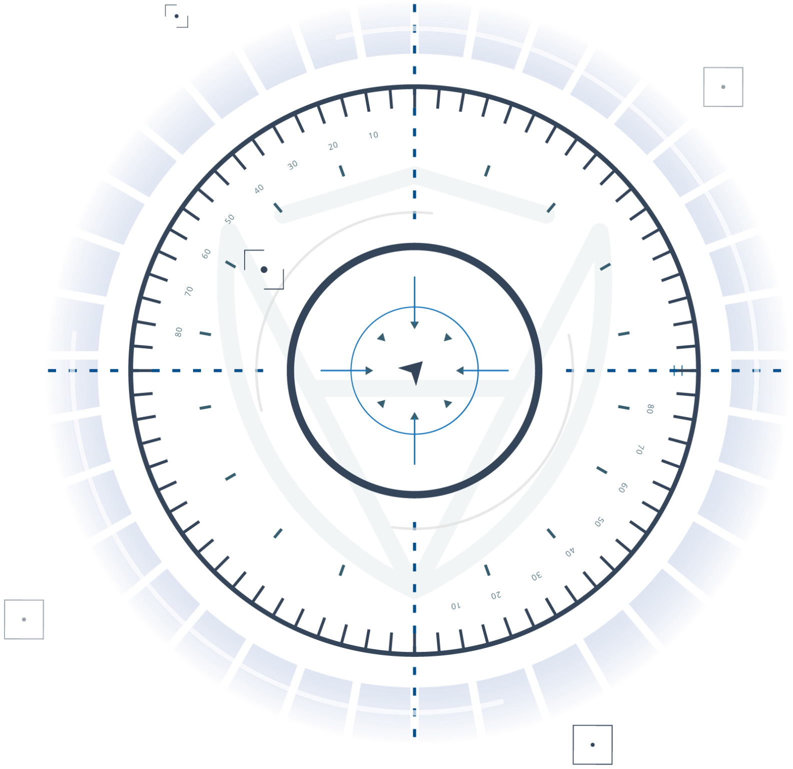
CERTIK
FOUNDATION

WaultFinance Protocol

Skynet Scanning Report

Feb 17th, 2021



Summary

This report is based on the results from CertiK Skynet Scanning, a proprietary security service that leverages automated scanning technologies to check smart contracts against a wide range of known vulnerabilities. Clients could reference the content to reason about the security score calculations. Automated static analysis can cover a wide range of known security issues and vulnerabilities, yet a full security assessment by security experts is always recommended to cover potential security concerns at business levels.

Scope of Works

The nature of CertiK Skynet is to provide real-time security intelligence, and scanning based on static analysis tool sets is one of the 6 Security Primitives. In summary:

1. Smart contracts run against CertiK's in-house tool chains (together with open source libraries for cross-checking purposes);
2. Manual efforts involved in reviewing the scanning outputs and filter out false alarms;
3. Manual efforts on a high level walkthrough of code logics to better understand the project and its intentions that may benefit a future full security assessment.

Contract Walkthroughs

WaultFinance

This token contract is not a typical `ERC20` implementation, but it fulfills the interface of `IERC20` and the majority of functions have introduced customized features that reflect the business logic. The source code is well structured and follows the common coding practices, yet we encourage more test cases and comments for security and readability considerations.

In summary, two major functionalities are introduced on top of the `ERC20` basics:

1. Fee: When transfer happens, fees may occur to different parties (as state variables) in certain situations. Fee rates are not mutable so could be considered as constants.
2. Reward: Similarly as Fee. When transfer happens, rewards may occur in certain situations. The reward rate is not constant, and is decreasing based on the ratio of rewardsSupply / actualSupply.

On the functional privileges, a total of 4 functions apply with `onlyOwner` modifier:

1. `excludeFromFees()`
2. `includeInFees()`
3. `excludeFromRewards()`
4. `includeInRewards()`

The owner has the control over who is able to receive the reward and who is exempt from the fee charged. When a user excludes from the reward system, all the rewards incurred will be converted to its balance.

WaultLiquidityMining

This liquidity mining contract is a typical DeFi scenario where users deposit LP tokens and receive `WAULT` tokens in return as rewards which would increase as the block height passes the starting point.

In summary, three core functions involved in state changes require special attention when interacting with:

1. `deposit()`: Callable by public and a user info record would be initiated or retrieved. A user-inputted amount of LP tokens would be deducted and in return logging the data that would be used to calculate the pending reward. The updatePool() would

be called at each function invocation to update the `accWaultPerShare` based on the delta of `block.number` and current balance of LP of the contract itself;

2. `withdraw()`: Callable by public and a user would provide an amount that would withdraw the LP token and the pending reward will be updated accordingly. A special indicator rewardDebt would also be updated that impacts the amount of rewards when claiming. The `updatePool()` would be called at each function invocation;
3. `claim()`: Callable by public and a user would receive the `WAULT` tokens based on the pending rewards and debts. Notice that if the pending reward amount is greater than the current balance of this contract's `WAULT`, then the latter one will be used as the amount transferring to the user. We envision that the `WAULT` team would constantly monitor the status of the contracts and fund in a timely manner. The `updatePool()` would be called at each function invocation.

On the functional privileges, a total of 3 functions apply with onlyOwner modifier:

1. `setWaultTokens()`
2. `startMining()`
3. `setWaultPerBlock()`

The owner has limited control over this contract, and the first two functions are considered as constructor functions that can only be called once. It's reasonable to see that the owner has the ability to adjust `WAULT` reward per block based on market situations and funding statuses.

WaultStaking

This contract is similar to the `WaultLiquidityMining` described above, and the major difference is that users need to stake WAULT tokens for a period of time (0, 7, 30 days) and get returns of `WAULT` tokens. The longer a user stakes and the higher yield would be processed. Require statements are in place to revert withdrawals that do not meet the staking timelines.

On the functional privileges, a total of 3 functions apply with onlyOwner modifier:

1. `setWaultToken()`
2. `startStaking()`
3. `setWaultPerBlock()`

Same to the explanation on `WaultLiquidityMining`. The owner has limited control over this contract, and the first two functions are considered as constructor functions that can only be called once. It’s reasonable to see that the owner has the ability to adjust `WAULT` reward per block based on market situations and funding statuses.

Scanning Results

Primitive Scores

Average Score | 98

- WaultFinance | 97
- WaultStaking | 97
- WaultLiquidityMining | 100

Smart Contracts

- WaultFinance
- WaultStaking
- WaultLiquidityMining

Source-code Primitive Results

In summary, 2 issues were found out of 33 checks for **WaultFinance**.

SWC-107	~~**-5 pts**~~
Title	Reentrancy
Contract	WaultFinance
Location	Line#66-67
Note	Consider false alarm

SWC-CTK-35	**-3 pts**
Title	Imprecise Arithmetic Operations Order
Contract	WaultFinance
Location	Line#250

In summary, 2 issues were found out of 33 checks for **WaultStaking**.

SWC-CTK-31	**~~-5 pts~~**
Title	Uninitialized State Variables
Contract	WaultStaking
Location	Line#34 & 71-83
Note	Consider false alarm, correct usage

SWC-CTK-35	**-3 pts**
Title	Imprecise Arithmetic Operations Order
Contract	WaultStaking
Location	Line#79, 80

In summary, 1 issue was found out of 33 checks for **WaultLiquidityMining**.

SWC-CTK-31	**~~-5 pts~~**
Title	Uninitialized State Variables
Contract	WaultLiquidityMining
Location	Line#30, 52-63
Note	Consider false alarm, correct usage

Disclaimer

Skynet Scanning could be leveraged as an automated toolset, however, it cannot replace a formal full security assessment, the toolset is best used synergistically alongside a full formal security audit. Security experts are extremely important in analyzing complex business logic and unknown vulnerabilities specific to each organization. QuickScan is a proprietary CertiK service, offered exclusively to existing and potential clients.

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services/verification, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

About CertiK

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. . Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

CERTIK
Provable Trust For All