DcentraLab Diligence

dcentralab.com/diligence

TokensFarm

Audit Report

TokensFarm

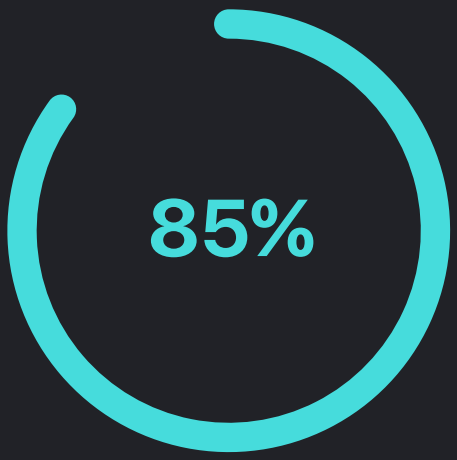
https://www.tokensfarm.com

Provided By DcentraLab Diligence on May 31, 2023

Security Audit Score

Pass

DcentraLab Diligence team has conducted an extensive audit on TokensFarm's Contracts and has found the code to be in low risk level given proper deployment and multi-sig permissioning



- Low Risk
- Small Risk
- Medium Risk
- High Risk

Scope

Audited Repository:

https://github.com/Tokensfarm/tokensfarm-contracts

Audited Branch:

develop

Audited Commit Hash:

51cdc995c9809b55716e9b971c0a39c391a94803

Contracts to be audited are the ones with visible changes in following diff:

ebe66d54336c85683cd919bbd26e94895a6d3b95...51cdc995c9809b55716e9b971c0a39c391a94803

* Latest develop vs latest master

Fix Commit Hash:

github.com/The-Poolz/DelayVault/commit/8a75232861aacc3862e4718e959792308511766a

Audited Contracts:

MerkleIterativeVesting.sol
MerkleLinearVesting.sol
TokensFarm.sol
TokensFarmSDK.sol
PerpetualTokensFarm.sol
PerpetualTokensFarmSDK.sol
TokensFarmFactory.sol
TokensFarmSDKFactory.sol
VestingFarmFactory.sol

Risks:

DcentraLab Diligence (DD) has performed all checks and verifications in its capacity to ascertain the safety of the code. However, it should be noted that misuse of the code, bad deployment practices, bad key management, exposing of private keys of the deployer and/or owner address and/or multi-sig signer addresses and/or fee collector address and/or any exposition of the code to malicious actors may result in an exploit of the code and loss of state and/or funds. Furthermore, there is always a chance that other Smart Contracts code could be written and deployed to cause the provided code by DD to act outside the intended scope by the client, to the point of causing state corruption or loss of funds to the client of the users of the code.

Issues Severity Reference Table

Type

Informational

This issue is not critical and does not pose an immediate threat to the functionality or security of the smart contract. It is simply an informational item that the auditors have identified and recommends addressing for best practices or to improve the overall performance of the contract.

Low

This issue is relatively minor and does not pose a significant risk to the functionality or security of the smart contract. While it is recommended to address these issues to ensure the highest level of quality and security, they are not likely to cause significant problems if left unaddressed.

Medium

This issue poses a moderate risk to the functionality or security of the smart contract. While it may not be immediately exploitable, it has the potential to cause problems in the future if left unaddressed. It is recommended to address these issues as soon as possible to prevent any potential negative impact on the contract.

High

This issue poses a significant risk to the functionality or security of the smart contract. Addressing these issues as soon as possible is recommended to prevent any potential negative impact on the contract. Failure to address these issues could result in significant problems and potential loss of funds or other assets.

Critical

This issue poses an immediate and severe risk to the functionality or security of the smart contract. It is recommended to address these issues immediately to prevent any potential negative impact on the contract. Failure to address these issues could result in catastrophic problems and significant loss of funds or other assets.

Discussion

The issue severity is dependent on design, centralization, and product specifications of the project.

Findings Summary

- Informational
- Low Risk
- Medium Risk
- High Risk
- Critical Risk
- Discussion

ID	Title	Severity	Status
1	Redundant uint value check	Low	Resolved
2	Missing merkle root diff check	Low	Acknowledged
3	Redundant storage reading	Informational	Unresolved
4	Redundant usage of SafeMath	Informational	Unresolved
5	Messy codebase	Informational	Unresolved
6	Missing error messages	Medium	Acknowledged
7	Improper function naming	Informational	Resolved
8	Unused function parameter	Low	Resolved
9	Unnecessary interface casting	Informational	Unresolved
10	Unnecessary declaration of a local variable	Informational	Acknowledged
11	Unnecessary storage reading	Low	Resolved

Findings Summary

ID	Title	Severity	Status
12	Unnecessary value setting	Low	Acknowledged
13	Redundant computation in _addPool() function	Low	Unresolved
14	Redundant global variable: noOfUsers	Low	Unresolved
15	Unoptimized if-else statement	Informational	Unresolved
16	Unoptimized for loop	Low	Unresolved
17	Redundant variable set to false	Informational	Resolved
18	Unoptimized memory readings in removeUsers function	Low	Resolved
19	Redundant else statement	Informational	Unresolved
20	Multiple interface castings	Informational	Unresolved
21	Breakable getter	Medium	Acknowledged
22	Bad naming in an event	Informational	Acknowledged
23	Redundant check and improper value setting	Low	Acknowledged
24	Informational - Redundant ‘using’ directive statement	Informational	Acknowledged
25	Unused Clones Library	Low	Acknowledged

Findings Summary

ID	Title	Severity	Status
26	Redundant value check	Informational	Resolved
27	Optimization runs	Informational	Acknowledged

Complete Analysis

General Notices:

There are multiple pieces of code repeating in between the contracts. Consider rechecking everything for the issues described in the single example.
*Consider adding annotations to the more complex flows as there is no way in which they can be properly analyzed without such.

ID 1: **Status: Resolved**

Low | Redundant uint value check

Present at: MerkleIterativeVesting @ L262 & @ L154-155 & @ L958

Description: Unsigned integers cannot, by definition, be of a value lesser than 0, so checking if a variable of uint type is greater than or equal to zero is redundant.

Recommendation: Remove the check.

Fix feedback: fixed as suggested

ID 2: **Status: Acknowledged**

Low | Missing merkle root diff check

Present at: MerkleIterativeVesting @ L254-L273

Description: At the end of the function marker isRootChanged is set to true, but the function flow does not contain a check that the new root is different than previous, therefore we're unable to conclude if root has been changed or not.

Complete Analysis

Recommendation: Introduce the check that confirms a new root is different from the previous.

Fix feedback: isRootChanged is just an indicator if setMerkleroot was called, root doesn't need to be actually changed. (Before isRootChanged was used differently where naming was compatible with logic now it is a little bit different, but we kept it the same because BE and FE were already built on top of it) - [Stayed the same]

ID 3: **Status: Unresolved**

Informational | Redundant storage reading

Present at: MerkleIterativeVesting @ L446

Description: While emitting an event there isActive variable is read from the storage which is unnecessary because this function sets state to false at all times.

Recommendation: Replace isActive with 'false' as the event argument.

Fix feedback: The event poller is tracking this event which is crucial for us to know when was emitted. Partial funding feature depends on this - [Stayed the same]

ID 4: **Status: Unresolved**

Informational | Redundant usage of SafeMath

Present at: MerkleIterativeVesting @ L1106

Description: Subtraction can be done without usage of SafeMath, because require statement above guarantees that contract balance is greater than pending rewards.

Recommendation: Use '-' instead of sub().

Fix feedback: Best practice to use safeMath everywhere [Stayed the same]

Complete Analysis

ID 5: **Status: Acknowledged**

Informational | Messy codebase

Present at: Throughout the repo

Recommendation: Apply prettier.

Fix feedback: [Stayed the same]

ID 6: **Status: Unresolved**

Medium | Missing error messages

Present at: Throughout the repo

Description: Require statements without error messages can lead to difficulties when debugging code and also leave users clueless (FE will not be able to write out the error message).

Recommendation: Add the error messages to require statements.

Fix feedback: All future features required me to prepare contract this way (Bytecode optimization) / Tenderly [Stayed the same]

Complete Analysis

ID 7: **Status: Resolved**

Informational | Improper function naming

Present at: MerkleIterativeVesting @ L874

Description: Function called hashData is not actually hashing but rather encoding the data.

Recommendation: Consider renaming function to make its use clearer.

Fix feedback: fixed as suggested

ID 8: **Status: Acknowledged**

Low | Unused function parameter

Present at: MerkleLinearVesting @ L544

Description: There's an unused parameter 'address user' in 'earned()' function.

Recommendation: Consider removing or commenting out the variable name to silence this warning.

Fix feedback: fixed as suggested

Complete Analysis

ID 9: **Status: Unresolved**

Informational | Unnecessary interface casting

Present at: PerpetualTokensFarm @ L501, L506

Description: Argument tokenAddress is parsed as 'address', but is not used in its original form (it is used only in casted form, as IERC20 token).

Recommendation: Consider changing function argument type from 'address' to 'IERC20' and remove additional casting.

Fix feedback: avoiding nonprimal types as params (Previous issues with verification) [Stayed the same]

ID 10: **Status: Acknowledged**

Informational | Unnecessary declaration of a local variable

Present at: PerpetualTokensFarm @ L513

Description: Local variable 'afterBalance' is declared but used only once.

Recommendation: Consider removing the variable and using data directly.

Fix feedback: readability in the first plan [Stayed the same]

Complete Analysis

ID 11: **Status: Resolved**

Low | Unnecessary storage reading

Present at: PerpetualTokensFarm @ L784-787 & @ L811-814 & @ L833-836 & @ L765

Description: Event arguments are read from storage (mapping), instead of using function arguments (local variables). Reading from mapping is more expensive and unnecessary.

Recommendation: Emit an event with local variables as arguments and evade loading from storage. Fixing this will have a fine impact on gas consumption in such functions and overall bytecode size.

Fix feedback: fixed as suggested

ID 12: **Status: Acknowledged**

Low | Unnecessary value setting

Present at: PerpetualTokensFarm @ L405-406 & TokensFarmSDK @ L282-283

Description: In the flow of '_addPool' function values of 'accERC20PerShare[epochId]' and 'totalDeposits[epochId]' are being set to zero. Zero is the default value of such mappings so setting is unnecessary.

Recommendation: Consider removing the mentioned part of flow.

Fix feedback: In case we decide to support multiple pools these lines of code are going to be useful [Stayed the same]

Complete Analysis

ID 13: **Status: Unresolved**

Low | Redundant computation in _addPool() function

Present at: PerpetualTokensFarm @ L399-401

Description: Function is being called only through initialization, this makes it callable only once and its separation from 'initialize()' function is probably unnecessary, anyhow the computation of '_lastRewardTime' is redundant due to condition in 'initialize()' function that guarantees that 'startTime' is greater or equal to 'block.timestamp'.

Recommendation: Remove the described computation and set 'block.timestamp' as 'lastRewardTime' for epoch.

Notice: Think about the desired behavior of '_addPool()' and if it is meant to be callable only once.

Fix feedback: It is not redundant startTime can be equal to block.timestamp [Stayed the same]

ID 14: **Status: Unresolved**

Low | Redundant global variable: noOfUsers

Present at: PerpetualTokensFarm @ L441 and throughout the contract

Description: Variable has been declared and interacted with without a need to do so. Number of participants per epoch can be retrieved via 'participants[epochId].length'.

Recommendation: Consider removing the redundant variable.

Fix feedback: It is not redundant, we don't have the function that is returning the whole array of participants FE or BE needs it for informational purposes [Stayed the same]

Complete Analysis

ID 15: **Status: Unresolved**

Informational | Unoptimized if-else statement

Present at: MerkleIterativeVesting @ L180-185

Description: Since default value of partialFunding is false there is no need to set it to false inside the if statement.

Recommendation: Consider checking if the initialFundPercent is less than 100 and then set the partialFunding to true (otherwise do not set it).

Fix feedback:There is no big upside of changing it (only risk of breaking code) [Stayed the same]

ID 16: **Status: Unresolved**

Low | Unoptimized for loop

Present at: MerkleIterativeVesting @ L188-198 & @ L297-307

Description: For loop is on each step checking if 'i == 0', since 'i' is 0 only in the first loop. This check is redundant for any other.

Recommendation: Consider executing logic from the if statement before looping and then start with 'i = 1'.

Fix feedback: There is no big upside of changing it (only risk of breaking code) [Stayed the same]

Complete Analysis

ID 17: **Status: Resolved**

Informational | Redundant variable set to false

Present at: MerkleIterativeVesting @ L201

Description: Variable which's default value is 'false' is being additionally set to 'false'.

Recommendation: Consider removing this statement.

Fix feedback: fixed as suggested

ID 18: **Status: Resolved**

Low | Unoptimized memory readings in removeUsers function

Present at: MerkleIterativeVesting @ L354-371 & MerkleLinearVesting @ L322-338

Description: Inside the function's for loop 'users[i]' has been accessed 10 times instead of once. This behavior increases gas consumption of this function significantly.

Recommendation: Consider instantiating the local variable 'address user = users[i]', and then use it throughout the loop.

Fix feedback: fixed as suggested

ID 19: **Status: Unresolved**

Informational | Redundant else statement

Present at: MerkleIterativeVesting @ L811

Description: Else statement present at described line is redundant as both statements return and therefore stop function flow in place.

Complete Analysis

Recommendation: Consider removing the else statement and just leave a second return outside of 'if'.

Fix feedback: real indicator that user is non existant is totalUserRewards[user] == 0 hence the if part [Stayed the same]

ID 20: **Status: Unresolved**

Informational | Multiple interface castings

Present at: MerkleIterativeVesting @ L819-844

Description: In multiple spots interface is being casted over the address.

Recommendation: Consider changing the flow in such a way that interface is casted only once.

Fix feedback: There is no space on the contract for additional global params [Stayed the same]

ID 21: **Status: Acknowledged**

Medium | Breakable getter

Present at: TokensFarmSDK @ L636-657

Description: Getter will break when the number of pendingSteaks reaches a certain threshold.

Recommendation: Consider changing the getter in such a manner that pending stakes are partially retrievable.

Fix feedback: practice showed that the user won't have more than 4 pending stakes which is far from breaking getter. Will take into consideration for future fixation [Stayed the same]

Complete Analysis

ID 22: **Status: Acknowledged**

Informational | Bad naming in an event

Present at: MerkleIterativeVesting @ L85

Description: 'UsersRemoved' event contains an argument called 'user' of type address[].

Recommendation: Consider changing the name of the argument to 'users'.

Fix feedback: Event poller already working with old event naming [Stayed the same]

ID 23: **Status: Acknowledged**

Low | Redundant check and improper value setting

Present at: TokensFarmSDK @ L713 & @ L1110

Description: Condition present at the mentioned line returns the lastRewardTime if it is less than endTime, since value of lastRewardTime can at most be equal to the endTime this ternary operator is redundant. This flow is not respected in a single place at line 1110, where 'lastRewardTime' is set to be 'block.timestamp' value (potentially greater than 'lastTime/endTime'). This does not disrupt the contract flow but it could lead to logical errors or misinformation in the future cases.

Recommendation: Discuss if this behavior is intentional or not, and make the changes based on discussion conclusions. If this is not a wanted behavior we recommend to delete the ternary operator, use 'lastRewardTime' as is, and set 'lastRewardTime' to equal 'lastTime' at L1110.

Fix feedback: Behaviour intentional [Stayed the same]

Complete Analysis

ID 24: **Status: Acknowledged**

Informational | Redundant 'using' directive statement

Present at: VestingFarmFactory @ L22 & TokensFarmFactory @ L20 & TokensFarmSDKFactory @ L20

Description: Clone library is usable only for type 'address'.

Recommendation: Consider rewriting the statement or removing it.

Fix feedback: It is not used but we will keep it there in case of usage in the future [Stayed the same]

ID 25: **Status: Acknowledged**

Low | Unused Clones Library

Present at: VestingFarmFactory & TokensFarmFactory & TokensFarmSDKFactory

Description: Clones library by OZ is imported and despite written 'using' directive has not been used inside the contracts.

Recommendation: Consider removing it.

Fix feedback: It is not used but we will keep it there in case of usage in the future [Stayed the same]

Complete Analysis

ID 26: **Status: Resolved**

Informational | Redundant value check

Present at: VestingFarmFactory @ L749 & TokensFarmFactory @ 1022 & TokensFarmSDKFactory @ 1043

Description: In lines described above there are checks of a uint value being greater than or equal to zero, which has no effect.

Recommendation: Consider removing the checks.

Fix feedback: fixed as suggested

ID 27: **Status: Acknowledged**

Informational | Optimization runs

Present at: Whole repository

Tip: Using optimization runs value above 200 can help you optimize bytecode for cheaper transactions and therefore better ux.

Fix feedback: [Stayed the same]

Disclaimer:

DcentraLab Diligence (DD) has provided the code to the client as is and assumes no responsibility nor legal liability for any use client may do with the code. Any and all usage and/or deployment of the code provided by DcentraLab Diligence will be done solely by the client, at the sole discretion, responsibility, risk, and legal liability of the Client, and DD will not be held accountable or liable for any loss of funds, security exploits or incidents, or any other unintended or negative outcome that may occur in relation to the code provided by DD.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts DD to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model, or legal compliance.

This report and the provided code or services as part of the SOW pertaining to this report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should it be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. DD's position is that each company and individual are responsible for their own due diligence and continuous security. DD's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

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

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, DcentraLab Diligence (DD) HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, DD SPECIFICALLY DISCLAIMS

ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, DD MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT / VERIFICATION REPORT, WORK PRODUCT, CODE OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

WITHOUT LIMITATION TO THE DISCLAIMER [ASSESSMENT NAME] FOREGOING, DD PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET THE CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR-FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER DD NOR ANY OF DD'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION, CODE OR CONTENT PROVIDED THROUGH THE SERVICE. DD WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT OR CODE, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, CODE, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS," AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN THE CUSTOMER AND THE THIRD-PARTY
OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS. THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO THE CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT DD'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST DD WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS. THE REPRESENTATIONS AND WARRANTIES OF DD CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF THE CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST DD WITH RESPECT TO SUCH
REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.
FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS, CODE, OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL,
REGULATORY, OR OTHER ADVICE.

dcentralab.com/diligence

DcentraLab Diligence

Provided By DcentraLab Diligence on May 31, 2023