CERTIK

Preliminary Comments

Tokensfarm #5

Mar 26th, 2022

Table of Contents

Summary

This report has been prepared for Tokensfarm #5 to discover issues and vulnerabilities in the source code of the Tokensfarm #5 project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- 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.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Tokensfarm #5
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/Tokensfarm/tokensfarm-contracts
Commit	9bd6786534954268cd57f0f7d1125ff25126e9e1

Audit Summary

Delivery Date	Mar 26, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

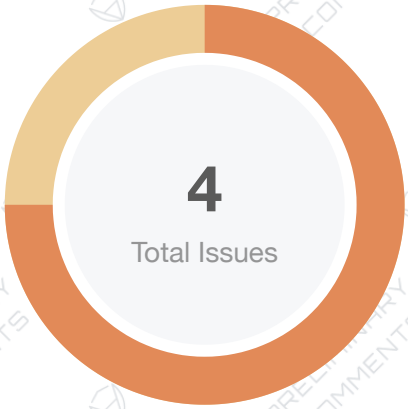
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	3	3	0	0	0	0	0
Medium	0	0	0	0	0	0	0
Minor	1	1	0	0	0	0	0
Informational	0	0	0	0	0	0	0
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
IVF	contracts/IterativeVestingFarm.sol	a1d45b706f498e0e7b8f974922979997f1030273b69d8fb3341bb9d92ae92733
LVF	contracts/LinearVestingFarm.sol	7babf9742614abc01d2545ede7e6546d98385ddd1bcfb65004c1778844473c3a
TFF	contracts/TokensFarmFactory.sol	be3ec4bf4d44ca10f2ce94ab6c512afeaeb4089b07a673f9055b21de6e544f13

Findings



Critical	**0** (0.00%)
Major	**3** (75.00%)
Medium	**0** (0.00%)
Minor	**1** (25.00%)
Informational	**0** (0.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
CON-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Pending
CON-02	Potential Front-Running Risk	Volatile Code	Minor	Pending
LVF-01	Incorrect `totalWithdrawn`	Logical Issue	Major	Pending
TFF-01	Centralized Control of Contract Upgrade	**Centralization / Privilege**	**Major**	Pending

CON-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	Major	contracts/IterativeVestingFarm.sol: 188, 227, 263, 290, 541, 576 contracts/LinearVestingFarm.sol: 141, 176, 212, 239, 254, 475, 510 contracts/TokensFarmFactory.sol: 275, 338, 408, 474, 498, 522, 544, 566, 589, 613, 638, 667, 695, 723, 752, 781, 808, 829, 854, 890, 912, 936, 960, 997, 1017, 1038, 1059, 1082, 1106, 1128, 1154, 1180, 1203	Pending

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the considerations of the administrator team's anonymity.

The `owner` of `IterativeVestingFarm` has the responsibility to notify users about the following capabilities:

- add users' rewards through `addUsersRewards()`
- remove user from farm through `removeUser()`
- pause the farm through `pauseFarm()`
- remove leftover rewards to the `collector` through `removeLeftOverRewards()`
- withdraw assets on the farm to the `collector` through `emergencyAssetsWithdrawal()`
- fund the farm and active through `fundAndOrActivate()`

The `owner` of `LinearVestingFarm` has the responsibility to notify users about the following capabilities:

- add users' rewards through `addUsersRewards()`
- remove user from farm through `removeUser()`
- pause the farm through `pauseFarm()`
- set the `endTime` through `setEndTime()`
- remove leftover rewards to the `collector` through `removeLeftOverRewards()`
- withdraw assets on the farm to the `collector` through `emergencyAssetsWithdrawal()`
- fund the farm and active through `fundAndOrActivate()`

Any compromise to the `owner` account may allow a hacker to take advantage of this authority.

The `maintainer` of `TokensFarmFactory` has the responsibility to notify users about the following capabilities:

- deploy and fund tokens farm through `deployAndFundTokensFarm()`
- deploy and fund linear vesting farm through `deployLinearVestingFarm()`

- deploy and fund iterative vesting farm through `deployIterativeVestingFarm()`
- fund again the tokens farm if necessary through `fundTheSpecificFarm()`
- fund again the linear vesting farm if necessary through `fundAndOrActivateSpecificLinearFarm()`
- fund again the iterative vesting farm if necessary through `fundAndOrActivateSpecificIterativeFarm()`
- pause the linear vesting farm through `pauseLinearSpecificFarm()`
- pause the iterative vesting farm through `pauseIterativeSpecificFarm()`
- add more users on linear vesting farm through `addMoreUsersOnSpecificLinearFarm()`
- add more users on iterative vesting farm through `addMoreUsersOnSpecificIterativeFarm()`
- set `minTimeToStake` in tokens farm through `setMinTimeToStakeOnSpecificFarm()`
- set `isEarlyWithdrawAllowed` in tokens farm through `setIsEarlyWithdrawAllowedOnSpecificFarm()`
- set `stakeFeePercent` in tokens farm through `setStakeFeePercentOnSpecificFarm()`
- set `rewardFeePercent` in tokens farm through `setRewardFeePercentOnSpecificFarm()`
- set `flatFeeAmount` in tokens farm through `setFlatFeeAmountOnSpecificFarm()`
- set `isFlatFeeAllowed` in tokens farm through `setIsFlatFeeAllowedOnSpecificFarm()`

Any compromise to the `maintainer` account may allow a hacker to take advantage of this authority.

The `tokensFarmCongress` of `TokensFarmFactory` has the responsibility to notify users about the following capabilities:

- remove users from the linear vesting farm through `removeUserOnSpecificLinearFarm()`
- remove users from the iterative vesting farm through `removeUserOnSpecificIterativeFarm()`
- withdraw the remaining funds left on the linear vesting farm through `withdrawLeftOverTokensOnSpecificLinearVestingFarm()`
- withdraw the remaining funds left on the iterative vesting farm through `withdrawLeftOverTokensOnSpecificIterativeVestingFarm()`
- withdraw assets on the linear vesting farm to the `feeCollector` through `emergencyAssetsWithdrawalOnSpecificLinearVestingFarm()`
- withdraw assets on the iterative vesting farm to the `feeCollector` through `emergencyAssetsWithdrawalOnSpecificIterativeVestingFarm()`
- withdraw fee collected in ERC value through `withdrawCollectedFeesERCOnSpecificFarm()`
- withdraw fee collected in ETH value through `withdrawCollectedFeesETHOnSpecificFarm()`
- withdraw stuck tokens on the farm through `withdrawTokensIfStuckOnSpecificFarm()`
- set `farmImplementation` through `setTokensFarmImplementation()`
- set `linearVestingFarmImplementation` through `setLinearVestingFarmImplementation()`
- set `iterativeVestingFarmImplementation` through `setIterativeVestingFarmImplementation()`

- set `farmImplementation`,`linearVestingFarmImplementation` and `iterativeVestingFarmImplementation` through `setAllImplementationAtOnce()`
- set `feeCollector` through `setFeeCollector()`
- set `feeCollector` in tokens farm through `setCurrentFeeCollectorOnSpecificFarm()`
- set `endTime` in linear vesting farm through `setEndTimeOnSpecificLinearVestingFarm()`
- set `proxyAdmin` through `setProxyAdmin()`

Any compromise to the `tokensFarmCongress` account may allow a hacker to take advantage of this authority.

Recommendation

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., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND

- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.

  OR
- Remove the risky functionality.

CON-02 | Potential Front-Running Risk

Category	Severity	Location	Status
Volatile Code	Minor	contracts/IterativeVestingFarm.sol: 109 contracts/LinearVestingFarm.sol: 92 contracts/TokensFarmFactory.sol: 71	Pending

Description

Malicious hackers may observe the pending transaction which will execute the `initialize` function, and launch a similar transaction but with the hacker's address of `owner` and gain the ownership of the contract.

Recommendation

We advise the client to design functionality to only allow a specific user to execute the `initialize` function.

LVF-01 | Incorrect `totalWithdrawn`

Category	Severity	Location	Status
Logical Issue	Major	contracts/LinearVestingFarm.sol: 433	Pending

Description

The `totalLeftLockedForUser` is the total remaining locked rewards, `claimAmountFromLocked` is the extractable rewards obtained based on the percentage of `totalLeftLockedForUser` and has been accumulated to `amountEarned` in line 420. Then `totalWithdrawn` should be the sum of `amountEarned` and `burnAmount`.

Recommendation

We advise the client to recheck the logic.

TFF-01 | Centralized Control Of Contract Upgrade

Category	Severity	Location	Status
Centralization / Privilege	Major	contracts/TokensFarmFactory.sol: 275, 338, 408	Pending

Description

The contract is an upgradeable contract, the proxy admin can upgrade the contract without the community's commitment. If an attacker compromises the account, he can change the implementation of the contract and drain tokens from the contract.

Recommendation

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.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;

  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.

  OR
- Remove the risky functionality.

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.

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.

Volatile Code

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

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

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, and terms and conditions provided to you ("Customer" or 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, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

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 CertiK 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 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 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK'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 CertiK is subject to dependencies and under continuing development. You agree that your access and/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 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, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET 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 FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET 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 CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK 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 (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, 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 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 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 CERTIK'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 CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF 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 CERTIK 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 OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

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.