



Security Assessment

Tokensfarm (new scope)

Dec 18th, 2021

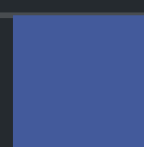


Table of Contents

Summary

Overview

[Project Summary](#)

[Audit Summary](#)

[Vulnerability Summary](#)

[Audit Scope](#)

Findings

[GLOBAL-01 : Potential Front-Running Risk](#)

[GLOBAL-02 : Logic Issue of Function `fundAndOrActivate\(\)`](#)

[GLOBAL-03 : Incorrect Withdrawn Amount](#)

[GLOBAL-04 : Lack of Zero Address Validation](#)

[GLOBAL-05 : Discussion For Function `addUsersRewards\(\)`](#)

[GLOBAL-06 : Function Visibility Optimization](#)

[GLOBAL-07 : Centralization Risk](#)

[GLOBAL-08 : Discussion For Function `removeUser\(\)`](#)

[IVF-01 : Potential Residual Rewards](#)

[IVF-02 : Discussion For `endTime != startTime` Condition Checking](#)

Appendix

Disclaimer

About

Summary

This report has been prepared for Tokensfarm (new scope) to discover issues and vulnerabilities in the source code of the Tokensfarm (new scope) 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 (new scope)
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/Tokensfarm/tokensfarm-contracts
Commit	019efb460c7883b759aabe19172a65bbe7b3acca 267c18689e61237476f1eb2c38ed43f524621afe

Audit Summary

Delivery Date	Dec 18, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

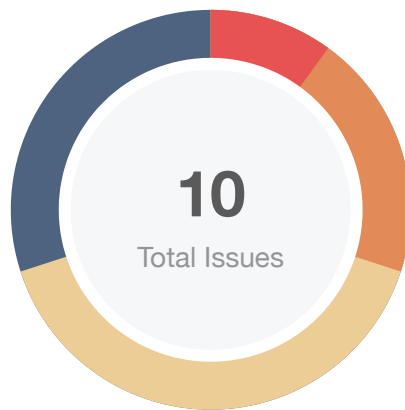
Vulnerability Summary

Vulnerability Level	Total	⚠ Pending	⊗ Declined	ℹ Acknowledged	🔄 Partially Resolved	✅ Resolved
🔴 Critical	1	0	0	0	0	1
🟠 Major	2	0	0	1	0	1
🟡 Medium	0	0	0	0	0	0
🟠 Minor	4	0	0	3	0	1
🟢 Informational	3	0	0	1	0	2
🟢 Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
IVF	IterativeVestingFarm.sol	22a72806b7575b54470fe2a2a47cde6c80dfb18517bd898b03b9345ae1c2007
LVF	LinearVestingFarm.sol	8d59d8ebde4616bf78fc7b91342e563c79790e9f2b6eb1088d9070cac1ff6b47
TFF	TokensFarmFactory.sol	52ddde28d0eb3e6c18ec2bfe409ee31b6169fe65305aabf05b80e93740ff8048

Findings



Critical	1 (10.00%)
Major	2 (20.00%)
Medium	0 (0.00%)
Minor	4 (40.00%)
Informational	3 (30.00%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Potential Front-Running Risk	Volatile Code	Minor	ⓘ Acknowledged
GLOBAL-02	Logic Issue of Function <code>fundAndOrActivate()</code>	Logical Issue	Major	✓ Resolved
GLOBAL-03	Incorrect Withdrawn Amount	Logical Issue	Critical	✓ Resolved
GLOBAL-04	Lack of Zero Address Validation	Volatile Code	Minor	ⓘ Acknowledged
GLOBAL-05	Discussion For Function <code>addUsersRewards()</code>	Logical Issue	Informational	✓ Resolved
GLOBAL-06	Function Visibility Optimization	Gas Optimization	Informational	✓ Resolved
GLOBAL-07	Centralization Risk	Centralization / Privilege	Major	ⓘ Acknowledged
GLOBAL-08	Discussion For Function <code>removeUser()</code>	Logical Issue	Informational	ⓘ Acknowledged
IVF-01	Potential Residual Rewards	Logical Issue	Minor	ⓘ Acknowledged
IVF-02	Discussion For <code>endTime != startTime</code> Condition Checking	Logical Issue	Minor	✓ Resolved

GLOBAL-01 | Potential Front-Running Risk

Category	Severity	Location	Status
Volatile Code	● Minor	Global	ⓘ Acknowledged

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.

For example:

- `LinearVestingFarm.initialize()`
- `IterativeVestingFarm.initialize()`
- `TokensFarmFactory.initialize()`

Recommendation

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

Alleviation

No alleviation.

GLOBAL-02 | Logic Issue of Function `fundAndOrActivate()`

Category	Severity	Location	Status
Logical Issue	● Major	Global	🟢 Resolved

Description

Anyone could invoke the function `fundAndOrActivate()`, which is to fund the farm and to set `isActive` to `true`. We would like to confirm with the client if the current implementation aligns with the original project design.

- `IterativeVestingFarm.fundAndOrActivate()`
- `LinearVestingFarm.fundAndOrActivate()`

Alleviation

The client resolved this issue by add the modifier `onlyOwner` in commit :
267c18689e61237476f1eb2c38ed43f524621afe.

GLOBAL-03 | Incorrect Withdrawn Amount

Category	Severity	Location	Status
Logical Issue	● Critical	Global	✓ Resolved

Description

The function `removeLeftOverRewards()` is used to transfer the leftover rewards to the `collector`. However, the function transferred all tokens of the contract instead of the remainder.

- `LinearVestingFarm.removeLeftOverRewards()`

```
341 vestedToken.safeTransfer(collector, vestedToken.balanceOf((address(this))));
```

- `IterativeVestingFarm.removeLeftOverRewards()`

```
359 vestedToken.safeTransfer(collector, vestedToken.balanceOf((address(this))));
```

Recommendation

We advise the client to recheck the logic.

Alleviation

The client resolved this issue in commit : 267c18689e61237476f1eb2c38ed43f524621afe.

GLOBAL-04 | Lack of Zero Address Validation

Category	Severity	Location	Status
Volatile Code	● Minor	Global	ⓘ Acknowledged

Description

The given input is missing the check for the non-zero address. For example:

- `LinearVestingFarm.removeLeftOverRewards()`
- `IterativeVestingFarm.removeLeftOverRewards()`

Recommendation

We advise the client to add the check for the passed-in values to prevent unexpected errors.

Alleviation

No alleviation.

GLOBAL-05 | Discussion For Function `addUsersRewards()`

Category	Severity	Location	Status
Logical Issue	● Informational	Global	✓ Resolved

Description

There is no validation of the `endTime`. If the farm is over, the following functions still work. If the `r.amount` is 0, this would cause the user to be added repeatedly in the next call to the function.

We would like to confirm with the client if the current implementation aligns with the original project design.

- `LinearVestingFarm.addUsersRewards()`
- `IterativeVestingFarm.addUsersRewards()`

Recommendation

The client revised the code and resolved this issue in commit :

267c18689e61237476f1eb2c38ed43f524621afe.

GLOBAL-06 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	● Informational	Global	🟢 Resolved

Description

`public` functions that are never called by the contract could be declared `external`. When the inputs are arrays, `external` functions are more efficient than `public` functions.

For example:

- `IterativeVestingFarm.withdraw()`
- `LinearVestingFarm.withdraw()`

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

Alleviation

The client heeded our advice and resolved this issue in commit :
267c18689e61237476f1eb2c38ed43f524621afe.

GLOBAL-07 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Major	Global	ⓘ Acknowledged

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()`

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 `deployAndFundLinearVestingFarm()`
- deploy and fund iterative vesting farm through `deployAndFundIterativeVestingFarm()`
- 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()`

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 `feeCollector` through `setFeeCollector()`
- set `feeCollector` in tokens farm through `setCurrentFeeCollectorOnSpecificFarm()`
- set `endTime` in linear vesting farm through `setEndTimeOnSpecificLinearVestingFarm()`

Recommendation

We advise the client to carefully manage the privileged account's private keys to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via 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 the different levels in terms of the short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

GLOBAL-08 | Discussion For Function `removeUser()`

Category	Severity	Location	Status
Logical Issue	● Informational	Global	ⓘ Acknowledged

Description

The function `addUsersRewards()` adds users into the `users` and updates the `userId`. However, the function `removeUser()` does not remove users from the `users` and delete the `userId` when the `totalUserRewards` is 0, this would cause the user to be added repeatedly in the next call to `addUsersRewards()`.

We would like to confirm with the client if the current implementation aligns with the original project design.

Alleviation

No alleviation.

IVF-01 | Potential Residual Rewards

Category	Severity	Location	Status
Logical Issue	Minor	projects/TokensFarm/contracts/IterativeVestingFarm.sol (1a44f85): 293, 313~316	① Acknowledged

Description

Performs a multiplication on the result of a division. Solidity integer division might truncate. As a result, performing multiplication before division can sometimes avoid loss of precision.

```
293  uint256 rewardPerPortion = (totalUserRewards[address(msg.sender)] /  
nVestingPortions);
```

```
313  amountEarned =(((nPortionsVested -  
userLastWithdrawnPortionNumber[address(msg.sender)]))  
314      * rewardPerPortion);
```

Recommendation

We advise the client to recheck the logic.

Alleviation

No alleviation.

IVF-02 | Discussion For `endTime != startTime` Condition Checking

Category	Severity	Location	Status
Logical Issue	● Minor	projects/TokensFarm/contracts/IterativeVestingFarm.sol (1a44f85): 203	✓ Resolved

Description

The `endTime != startTime` validation only exists in the function `removeUser()`.

We would like to know why there is such a difference.

Alleviation

The client resolved this issue by removing the logic in commit :

267c18689e61237476f1eb2c38ed43f524621afe.

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

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.

