Security Assessment

Plearn

Nov 30th, 2021

Table of Contents

Summary

This report has been prepared for Plearn to discover issues and vulnerabilities in the source code of the Plearn 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	Plearn
Platform	BSC
Language	Solidity
Codebase	https://github.com/plearnclub/plearn-swap-core https://github.com/plearnclub/plearn-swap-periphery https://github.com/plearnclub/plearn-farm
Commit	https://github.com/plearnclub/plearn-swap-core/tree/8b9a35d067194eea1b3e61e1cbedd5f0e4462c1f https://github.com/plearnclub/plearn-swap-periphery/tree/5f00b20fb1c9271839f5c6ecf41af9c41578c8cb https://github.com/plearnclub/plearn-farm/tree/eec25499bbb8be3eecba61c03572974845ee1ef7 https://github.com/plearnclub/plearn-swap-periphery/tree/023331b95e7673bcea1e3286e14f74b6f66335fd https://github.com/plearnclub/plearn-farm/tree/476c1b4b1dfa86f9353595690ef30a2f2aa1a1ce

Audit Summary

Delivery Date	Nov 30, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

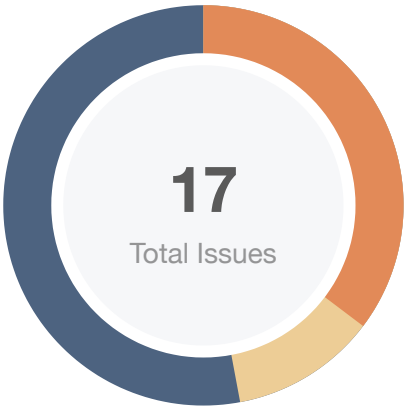
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	6	0	0	6	0	0
Medium	0	0	0	0	0	0
Minor	2	0	0	0	0	2
Informational	9	0	0	0	0	9
Discussion	0	0	0	0	0	0

Audit Scope

ID			File	SHA256 Checksum

Findings



Critical	**0** (0.00%)
Major	**6** (35.29%)
Medium	**0** (0.00%)
Minor	**2** (11.76%)
Informational	**9** (52.94%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
MCC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
PLC-01	Changes Not Listed In Change List	Inconsistency	Informational	Resolved
PRC-01	Changes Not Listed In Change List	Inconsistency	Informational	Resolved
PRK-01	Changes Not Listed In Change List	Inconsistency	Informational	Resolved
PTC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
PTC-02	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
SCC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
SCC-02	Storage Manipulation In `view` Functions	Gas Optimization	Informational	Resolved
SCC-03	Missing Input Validation	Volatile Code	Informational	Resolved
SCC-04	Missing Emit Events	Coding Style	Informational	Resolved
SCF-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
SCI-01	Inconsistent Implementation Between `deposit()` And `depositToInvestor()`	Logical Issue	Minor	Resolved

ID	Title	Category	Severity	Status
SCI-02	Storage Manipulation In `view` Functions	Gas Optimization	Informational	Resolved
SCI-03	Missing Input Validation	Volatile Code	Informational	Resolved
SCI-04	Missing Emit Events	Coding Style	Informational	Resolved
TCC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
TCC-02	Potential Reentrancy Risks	Logical Issue	Minor	Resolved

MCC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/plearn-farm/contracts/MasterChef.sol (501dcff): 418	Acknowledged

Description

In the contract `PlearnToken`, the role `Owner` has the authority over the following function:

- addMinter
- delMinter
- getMinter

In the contract `MasterChef`, the role `Owner` has the authority over the following function:

- setDevAddress
- setRefAddress
- setSafuAddress
- updatePlearnPerBlock

Any compromise to the `Owner` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `Owner` account's private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

PLC-01 | Changes Not Listed In Change List

Category	Severity	Location	Status
Inconsistency	Informational	projects/plearn-swap-periphery/contracts/libraries/PlearnLibrary.sol (501dcff): 125, 110, 95, 90, 79, 75	Resolved

Description

The linked code change is different from the pancake swap contracts but not specified in the changes list provided by the Plearn team:

- Add new input variable `swapFee` in `getAmountIn` and `getAmountOut`

Alleviation

[Plearn Team]: Update the changes list.

PRC-01 | Changes Not Listed In Change List

Category	Severity	Location	Status
Inconsistency	Informational	projects/plearn-swap-periphery/contracts/PlearnRouter02.sol (501dcff): 489, 487, 480, 478	Resolved

Description

The linked code change is different from the pancake swap contracts but not specified in the changes list provided by the Plearn team:

- Add new input variable `swapFee` in `getAmountIn` and `getAmountOut`

Alleviation

[Plearn Team]: Update the changes list.

PRK-01 | Changes Not Listed In Change List

Category	Severity	Location	Status
Inconsistency	Informational	projects/plearn-swap-periphery/contracts/PlearnRouter01.sol (501dcff): 348, 346, 339, 337	Resolved

Description

The linked code change is different from the pancake swap contracts but not specified in the changes list provided by the Plearn team:

- Add new input variable `swapFee` in `getAmountIn` and `getAmountOut`

Alleviation

[Plearn Team]: Update the changes list.

PTC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/plearn-farm/contracts/PlearnToken.sol (501dcff): 306, 290, 282	Acknowledged

Description

In the contract `PlearnToken`, the role `Owner` has the authority over the following function:

- addMinter
- delMinter
- getMinter

In the contract `MasterChef`, the role `Owner` has the authority over the following function:

- setDevAddress
- setRefAddress
- setSafuAddress
- updatePlearnPerBlock

Any compromise to the `Owner` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `Owner` account's private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

PTC-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/plearn-farm/contracts/PlearnToken.sol (501dcff): 19	Acknowledged

Description

In the contract `PlearnToken`, the role `Minter` has the authority over the following function:

- mint

Any compromise to the `Minter` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `Minter` account's private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

SCC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/plearn-farm/contracts/SmartChef.sol (501dcff): 238, 226, 209, 186, 199, 176	Acknowledged

Description

In the contract `SmartChef`, the role `owner` has the authority over the following function:

- emergencyRewardWithdraw
- recoverWrongTokens
- stopReward
- updatePoolLimitPerUser
- updateRewardPerBlock
- updateStartAndEndBlocks

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

Recommendation

We advise the client to carefully manage the `Owner` account's private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

SCC-02 | Storage Manipulation In `view` Functions

Category	Severity	Location	Status
Gas Optimization	Informational	projects/plearn-farm/contracts/SmartChef.sol (501dcff): 258	Resolved

Description

There should not be any storage variable manipulation in the `view` function.

Recommendation

We advise the client to consider changing `storage` into `memory`.

Alleviation

[Plearn Team]: changed the linked variable declaring from storage to memory.

SCC-03 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	projects/plearn-farm/contracts/SmartChef.sol (501dcff): 82~83	Resolved

Description

The given input is missing the check for `bonusEndBlock` should larger than `startBlock` as the restriction as updating:

```
239         require(block.number < startBlock, "Pool has started");
240         require(_startBlock < _bonusEndBlock, "New startBlock must be lower than new
endBlock");
241         require(block.number < _startBlock, "New startBlock must be higher than
current block");
```

`startUnlockBlock` and `endUnlockBlock` are unchecked as the same.

Recommendation

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

Alleviation

[Plearn Team]: Add

```
require _startBlock < _bonusEndBlock
require block.number < _startBlock
require _startUnlockBlock < _endUnlockBlock
require block.number < _startUnlockBlock
```

to contructor() and function updateStartUnlockAndEndUnlockBlocks()

SCC-04 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	projects/plearn-farm/contracts/SmartChef.sol (501dcff): 60	Resolved

Description

In `SmartChef` and `smartChefFoundingInvestor`, the event `RewardsStop` is not emitted in function `stopReward`.

Recommendation

We advise the client to emit this event.

Alleviation

[Plearn Team]: Add the event emit in the stopReward function.

SCF-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/plearn-farm/contracts/SmartChefFoundingInvestorTreasury.sol (501dcff): 55, 39, 31, 22	Acknowledged

Description

In the contract `SmartChefFoundingInvestorTreasury`, the role `Owner` has the authority over the following function:

- addAdmin
- delAdmin
- getAdmin

In the contract `SmartChefFoundingInvestorTreasury`, the role `Admin` has the authority over the following function:

- safeRewardTransfer

Any compromise to the `Owner` or `Admin` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `Owner` or `Admin` accounts' private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

SCI-01 | Inconsistent Implementation Between `deposit()` And `depositToInvestor()`

Category	Severity	Location	Status
Logical Issue	Minor	projects/plearn-farm/contracts/SmartChefFoundingInvestor.sol (501dcff): 161~167, 149~151, 136, 127	Resolved

Description

There are inconsistent between function deposit and depositToInvestor:

1. Deposit amount within function `depositToInvestor()` follows the poolLimitPerUser validation, but `deposit` not.
2. There are no amount transfers from the caller and staked amount increase in `deposit`.

Recommendation

It's recommended to:

1. Follow the consistent logic in `deposit` and `depositToInvestor` if the client wants the investor could deposit by themselves.
2. Add the `require` `amount == 0` with the meaningful descriptions in `deposit` to expose the project's feature to their investors.

Alleviation

[Plearn Team]: refactor function name deposit() to harvest()

SCI-02 | Storage Manipulation In `view` Functions

Category	Severity	Location	Status
Gas Optimization	Informational	projects/plearn-farm/contracts/SmartChefFoundingInvestor.sol (501dcff): 323, 304	Resolved

Description

There should not be any storage variable manipulation in the `view` function.

Recommendation

We advise the client to consider changing `storage` into `memory`.

Alleviation

[Plearn Team]: changed the linked variable declaring from storage to memory.

SCI-03 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	projects/plearn-farm/contracts/SmartChefFoundingInvestor.sol (501dcff): 103~104, 101~102	Resolved

Description

The given input is missing the check for `bonusEndBlock` should larger than `startBlock` as the restriction as updating:

```
 239         require(block.number < startBlock, "Pool has started");
 240         require(_startBlock < _bonusEndBlock, "New startBlock must be lower than new
endBlock");
 241         require(block.number < _startBlock, "New startBlock must be higher than
current block");
```

`startUnlockBlock` and `endUnlockBlock` are unchecked as the same.

Recommendation

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

Alleviation

[Plearn Team]: Add

```
require _startBlock < _bonusEndBlock
require block.number < _startBlock
require _startUnlockBlock < _endUnlockBlock
require block.number < _startUnlockBlock
```

to contructor() and function updateStartUnlockAndEndUnlockBlocks()

SCI-04 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	projects/plearn-farm/contracts/SmartChefFoundingInvestor.sol (501dcff) : 74	Resolved

Description

In `SmartChef` and `smartChefFoundingInvestor`, the event `RewardsStop` is not emitted in function `stopReward`.

Recommendation

We advise the client to emit this event.

Alleviation

[Plearn Team]: Add the event emit in the stopReward function.

TCC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/plearn-farm/contracts/TimelockController.sol (501dcff): 567, 551, 535, 496, 469	Acknowledged

Description

In the contract `TimeLockController`, the role `PROPOSER_ROLE` has the authority over the following function:

- schedule
- scheduleBatch
- cancel

The role `EXECUTOR_ROLE` has the authority over the following function:

- execute
- executeBatch

Any compromise to the `PROPOSER_ROLE` or `EXECUTOR_ROLE` account may allow the hacker to take advantage of this if the locking time is not long enough.

Recommendation

We advise the client to carefully manage the `PROPOSER_ROLE` or `EXECUTOR_ROLE` accounts' private key 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 smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of 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

[Plearn Team]: manage the Owner account's private key

TCC-02 | Potential Reentrancy Risks

Category	Severity	Location	Status
Logical Issue	Minor	projects/plearn-farm/contracts/TimelockController.sol (501dcff): 573~576, 554~555	Resolved

Description

The function `execute` and `executeBatch` is risky to reentry attack. The function call eventually triggers the `target.call{value: value}(data)` in function `_call` and state variable `_timestamp[id]` may be changed later in the function `_afterCall`. Since the real implementation of the external contract is unclear, and the address behind the interface is not clear, reentrancy is possible to take place.

Recommendation

We advise the client to use the Checks-Effects-Interactions Pattern to avoid the risk of calling unknown contracts.

Alleviation

[Plearn Team]: add openzepplin ReentrancyGuard in function execute, executeBatch

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.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

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.