Preliminary Comments

Lawblocks

May 12th, 2022

Table of Contents

Summary

Overview

Project Summary

Audit Summary

Vulnerability Summary

Audit Scope

Findings

GLOBAL-01 : Centralization Risks in Lawblocks.sol

LLC-01 : Initial Token Distribution

LLC-02 : Incorrect `_totalSupply` Calculation

LLC-03 : `UpgradeableToken()` Function Unrestricted

LLC-04 : Inconsistent `_owner`

LLC-05 : Missing Zero Address Validation

LLC-06 : Deprecated Constructor Naming

LLC-07 : Missing Emit Events

LLC-08 : Improper Usage of `public` and `external` Type

LLC-09 : Missing Error Messages

LLC-10 : Unlocked Compiler Version

LLC-11 : Unimplemented Function

LLC-12 : Too Low `_cap`

LLC-13 : Unclear Usage of `Upgradeable`

Appendix

Disclaimer

About

Summary

This report has been prepared for Lawblocks to discover issues and vulnerabilities in the source code of the Lawblocks 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	Lawblocks
Platform	Ethereum
Language	Solidity
Codebase	https://xdc.network/token/xdc05940b2df33d6371201e7ae099ced4c363855dfe
Commit	

Audit Summary

Delivery Date	May 12, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

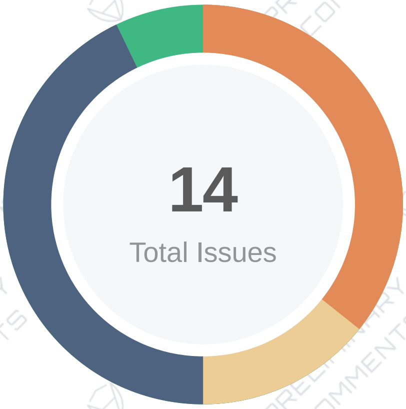
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	5	5	0	0	0	0	0
● Medium	0	0	0	0	0	0	0
● Minor	2	2	0	0	0	0	0
● Informational	6	6	0	0	0	0	0
● Discussion	1	1	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
LLC	projects/Lawblocks/contracts/Lawblocks.sol	7a3bbd9bcfde718e670e97bb5c5a97697107619727879894 5ccc9ef6dd44a83f

Findings



🟥	**Critical**	**0** (0.00%)
🟧	**Major**	**5** (35.71%)
🟨	**Medium**	**0** (0.00%)
🟨	**Minor**	**2** (14.29%)
🟦	**Informational**	**6** (42.86%)
🟩	**Discussion**	**1** (7.14%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Risks In Lawblocks.sol	**Centralization / Privilege**	● **Major**	Pending
LLC-01	Initial Token Distribution	**Centralization / Privilege**	● **Major**	Pending
LLC-02	Incorrect `_totalSupply` Calculation	Mathematical Operations	● Major	Pending
LLC-03	`UpgradeableToken()` Function Unrestricted	Control Flow	● Major	Pending
LLC-04	Inconsistent `_owner`	Inconsistency	● Major	Pending
LLC-05	Missing Zero Address Validation	Volatile Code	● Minor	Pending
LLC-06	Deprecated Constructor Naming	Language Specific	● Minor	Pending
LLC-07	Missing Emit Events	Coding Style	● Informational	Pending
LLC-08	Improper Usage Of `public` And `external` Type	Gas Optimization	● Informational	Pending
LLC-09	Missing Error Messages	Coding Style	● Informational	Pending
LLC-10	Unlocked Compiler Version	Language Specific	● Informational	Pending
LLC-11	Unimplemented Function	Compiler Error	● Informational	Pending
LLC-12	Too Low `_cap`	Logical Issue	● Informational	Pending

ID	Title	Category	Severity	Status
LLC-13	Unclear Usage Of `Upgradeable`	Logical Issue	● Discussion	ⓘ Pending

GLOBAL-01 | Centralization Risks In Lawblocks.sol

Category	Severity	Location	Status
Centralization / Privilege	● **Major**		ⓘ Pending

Description

In the contract `ERC20` the role `_owner` has authority over the functions below.

- `destroyToken`
- `sendTokensToOwner`
- `sendTokensToCrowdsale`

In the contract `Ownable` the role `_owner` has authority over the functions below.

- `transferOwnership`
- `renounceOwnership`

In the contract `Ownable` the role `newOwner` has authority over the functions below.

- `acceptOwnership`

In the contract `Upgradeable` the role `upgradeMaster` has authority over the functions below.

- `setUpgradeAgent`
- `setUpgradeMaster`

In the contract `CapperRole` the role `cappers` has authority over the functions below.

- `addCapper`

In the contract `SignerRole` the role `signers` has authority over the functions below.

- `addSigner`

In the contract `PauserRole` the role `pausers` has authority over the functions below.

- `addPauser`

In the contract `Pausable` the role `pausers` has authority over the functions below.

- `pause`

- `unpause`

In the contract `MinterRole` the role `minters` has authority over the functions below.

- `addMinter`

In the contract `ERC20Mintable` the role `minters` has authority over the functions below.

- `mint`
- `finishMinting`

Any compromise to privileged accounts may allow the hacker to take advantage of this authority and modify the contract configuration.

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.

LLC-01 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	projects/Lawblocks/contracts/Lawblocks.sol: 1114	ⓘ Pending

Description

All of the `LBT` tokens are sent to the contract address when deploying the contract. This could be a centralization risk as the owner can distribute `LBT` tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

LLC-02 | Incorrect `_totalSupply` Calculation

Category	Severity	Location	Status
Mathematical Operations	● Major	projects/Lawblocks/contracts/Lawblocks.sol: 788	Pending

Description

When token holders upgrade some of their tokens to a new contract.

```
      _balances[msg.sender] = _balances[msg.sender].sub(value);

      // Take tokens out from circulation
      _totalSupply = _totalSupply.add(value);
```

Recommendation

When upgrading, the total supply of tokens decreases, so we propose to modify it as follows.

```
     _totalSupply = _totalSupply.sub(value);
```

LLC-03 | `UpgradeableToken()` Function Unrestricted

Category	Severity	Location	Status
Control Flow	● Major	projects/Lawblocks/contracts/Lawblocks.sol: 770	ⓘ Pending

Description

Anyone can invoke the "UpgradeableToken" function.

```
function UpgradeableToken(address _upgradeMaster) {
  upgradeMaster = _upgradeMaster;
}
```

Recommendation

We recommend using the modifier `onlyOwner` to make the following changes to it.

```
function UpgradeableToken(address _upgradeMaster) onlyOwner{
```

LLC-04 | Inconsistent `_owner`

Category	Severity	Location	Status
Inconsistency	● Major	projects/Lawblocks/contracts/Lawblocks.sol: 287~292	Pending

Description

If a transfer of ownership occurs, the `onlyOwner` in ERC20 uses the new owner, but the `_owner` in the `_balance[_owner]` in the funtion is still the old one, resulting in `_tokens` being transferred to the deployer of the contract. .

```
    function sendTokensToOwner(uint _tokens) onlyOwner returns (bool ok){
      require(_balances[this] >= _tokens);
      _balances[this] =_balances[this].sub(_tokens);
      _balances[_owner] =_balances[_owner].add(_tokens);
      return true;
    }
```

Recommendation

We recommend rechecking the logic here to make sure it is correct.

LLC-05 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	● Minor	projects/Lawblocks/contracts/Lawblocks.sol: 771	ⓘ Pending

Description

Addresses should be checked before assignment or external call to make sure they are not zero addresses.

File: projects/Lawblocks/contracts/Lawblocks.sol (Line 771, Function `Upgradeable.UpgradeableToken`)

```
upgradeMaster = _upgradeMaster;
```

- `_upgradeMaster` is not zero-checked before being used.

Recommendation

We advise adding a zero-check for the passed-in address value to prevent unexpected errors.

LLC-06 | Deprecated Constructor Naming

Category	Severity	Location	Status
Language Specific	● Minor	projects/Lawblocks/contracts/Lawblocks.sol: 135~139	Pending

Description

In Solidity version 0.4.24 declaring a function with the contracts name as constructor has been depreciated in favor for constructor().

```
function ERC20(uint256 _value){
    _totalSupply = _value;
    _balances[this]= _totalSupply;
    _owner = msg.sender;
}
```

Recommendation

We recommend using a newer version and use the keyword constructor() to initialize the contract.

```
constructor(uint256 _value)public{
    _totalSupply = _value;
    _balances[this]= _totalSupply;
    _owner = msg.sender;
}
```

LLC-07 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 280, 287, 294, 836	ⓘ Pending

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

LLC-08 | Improper Usage Of `public` And `external` Type

Category	Severity	Location	Status
Gas Optimization	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 135, 153, 163, 179, 198, 212, 240, 264, 280, 287, 473, 573, 612, 651, 690, 770, 777, 836, 914, 925, 937, 948, 959, 983, 992, 1037, 1097	ⓘ Pending

Description

`public` functions that are never called by the contract could be declared as `external`. `external` functions are more efficient than `public` functions.

Recommendation

Consider using the external attribute for public functions that are never called within the contract.

LLC-09 | Missing Error Messages

Category	Severity	Location	Status
Coding Style	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 23, 32, 43, 54, 64, 128, 180, 181, 199, 220, 221, 222, 247, 271, 281, 288, 295, 309, 322, 323, 338, 367, 378, 388, 465, 474, 482, 523, 531, 544, 565, 604, 643, 682, 780, 783, 803, 805, 807, 809, 814, 816, 837, 838, 876, 884, 1020, 1080, 1104	ⓘ Pending

Description

The **require** can be used to check for conditions and throw an exception if the condition is not met. It is better to provide a string message containing details about the error that will be passed back to the caller.

Recommendation

We advise adding error messages to the linked **require** statements.

LLC-10 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 1	Pending

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.6.2` the contract should contain the following line:

```
pragma solidity 0.6.2;
```

LLC-11 | Unimplemented Function

Category	Severity	Location	Status
Compiler Error	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 725	ⓘ Pending

Description

These functions were not implemented within the scope of the contract for this audit.

File: projects/Lawblocks/contracts/Lawblocks.sol (Line 725, Contract `UpgradeAgent`)

```
function upgradeFrom(address _tokenHolder, uint256 _amount) external;
```

Recommendation

Please implement all unimplemented functions in any contract you intend to use directly (not simply inherit from).

LLC-12 | Too Low `_cap`

Category	Severity	Location	Status
Logical Issue	● Informational	projects/Lawblocks/contracts/Lawblocks.sol: 1114	ⓘ Pending

Description

We see that the contract deployment agrees on a `_cap` value of `10000000000000000000000000`, which is less than the number of initial tokens, so tokens cannot be minted after deployment.

Recommendation

Please review the `_cap` defined here.

LLC-13 | Unclear Usage Of `Upgradeable`

Category	Severity	Location	Status
Logical Issue	● Discussion	projects/Lawblocks/contracts/Lawblocks.sol: 732	ⓘ Pending

Description

It is not clear how `Upgradeable` is used in the current contract, it looks like be designed so that the tokens of this contract can be upgraded to various new tokens

In addition, defining the functions `canUpgrade` and `isUpgradeAgent` is not necessary, because they always return `true`.

Recommendation

Please introduce the usage of `Upgradeable`.

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.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

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.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

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

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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.

Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

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.