CERTIK

Security Assessment

TARS Protocol

Apr 30th, 2022

Table of Contents

Summary

This report has been prepared for TARS Protocol to discover issues and vulnerabilities in the source code of the TARS Protocol 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.

Additionally, this audit is based on a premise that all external contracts were implemented safely.

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	TARS Protocol
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/Tars-Protocol/tars_protocol
Commit	c9573fa39115aa2873c08f6ab5e13c1428a57259

Audit Summary

Delivery Date	Apr 30, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

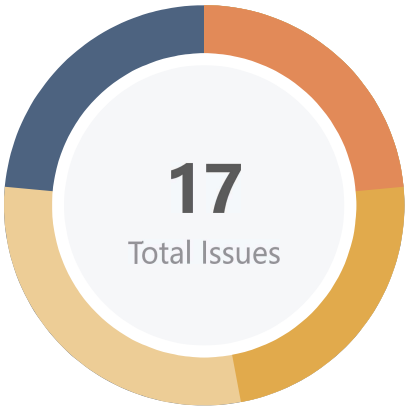
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	4	0	0	1	0	0	3
Medium	4	0	0	1	0	0	3
Minor	5	0	0	3	0	1	1
Informational	4	0	0	3	0	0	1
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
CSI	CoinSalesIDO.sol	cd8d2415a75364d4473c7774403ee697cf6b575094861dc53ebc5d64bb772eee
ALT	AssemblyLine.sol	52b0b2ab0e134dad04dfab7834b827f340fc8b03b9c5e4b5bf3b0ff06f81168e
VNT	token/VoucherNft.sol	d6f6f43ab261c2986193396dc988353001280a002b730f67998ccca341470247
FID	FactoryIDO.sol	7a5dec9eae43ca4dcf43de9ac53397e9f6ad143847507a228b9be08ba5475556
ITP	Insurance.sol	75c2b0a6ff5cd277083e605249fc4973dcb18c8da4961edfa05cb53a1c796cc0

Findings



Critical	**0** (0.00%)
Major	**4** (23.53%)
Medium	**4** (23.53%)
Minor	**5** (29.41%)
Informational	**4** (23.53%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
CSI-01	Missing Validation	Logical Issue	Major	Resolved
CSI-02	Missing Input Validation	Volatile Code	Minor	Resolved
CSI-03	Lack Of Reasonable Boundary	Volatile Code	Minor	Partially Resolved
CSI-04	Missing Input Validation	Volatile Code	Minor	Acknowledged
CSI-05	LockWarehouse Can Be Added After User Investment	Logical Issue	Minor	Acknowledged
CSI-06	Condition Can Be Simplified	Volatile Code	Informational	Resolved
FID-01	Incorrect Logic In Function `customProductionIDO`	Logical Issue	Medium	Acknowledged
ITP-01	Possible Loss Of Accuracy In Function `joinInsurance`	Logical Issue	Medium	Resolved
ITP-02	Incorrect Operator Used	Logical Issue	Medium	Resolved
ITP-03	Inadequate Validation	Logical Issue	Medium	Resolved
TPC-01	Permission Issues For Super Admin	Logical Issue	Major	Resolved
TPC-02	Missing Input Validation	Volatile Code	Minor	Acknowledged

ID	Title	Category	Severity	Status
TPC-03	Unlocked Compiler Version	Language Specific	Informational	Acknowledged
TPC-04	Missing Emit Events	Coding Style	Informational	Acknowledged
TPC-05	Variables That Could Be Declared As Immutable	Gas Optimization	Informational	Acknowledged
VNT-01	Missing Creation In Function `queryUserAllNft`	Language Specific	Major	Resolved

GLOBAL-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	**Global**	Acknowledged

Description

In the contract `CoinSalesIDO`, the role `superAdmin` has the authority over the following function:

1. function adminSafeTransfer()
2. function updateInsuranceAddress()
3. function updateIsPrivate()
4. function updateTime()
5. function updatePoolQuota()
6. function updateQuantitySold()
7. function updateWhiteListQuota()
8. function addWhiteList()
9. function addLockWarehouse()
10. function updateSellingToken()
11. function updateSellingTokenName()
12. function cancelProject()
13. function identification()
14. function transferCoin()

In the contract `CoinSalesIDO`, the role `secondParty` has the authority over the following function:

1. function addWhiteList()
2. function addLockWarehouse()
3. function updateSellingToken()
4. function updateSellingTokenName()
5. function cancelProject()
6. function secondPartyClaim()

In the contract `CoinSalesIDO`, the role `firstParty` has the authority over the following function:

1. function identification()
2. function transferCoin()
3. function firstPartyClaim()

In the contract `FactoryIDO`, the role `superAdmin` has the authority over the following function:

1. function adminSafeTransfer()
2. function updateAdmin()
3. function updateInsurance()
4. function updateAssemblyLine()
5. function addAssemblyLine()
6. function deleteAssemblyLine()

In the contract `FactoryIDO`, the role `voucherBoss` has the authority over the following function:

1. function updateProxyInvestment()
2. function foundCoinSalesIdo()
3. function customProductionIDO()

In the contract `FactoryIDO`, the role `operator` has the authority over the following function:

1. function updatePayee()
2. function updateBusinessToken()
3. function updatePrice()
4. function updateClosePrice()
5. function addFundRaisingToken()
6. function deleteFundRaisingToken()
7. function poolAuthentication()

In the contract `Insurance`, the role `superAdmin` has the authority over the following function:

1. function adminSafeTransfer()
2. function updateAdmin()
3. function setFactoryIdoAddress()

In the contract `Insurance`, the role `operator` has the authority over the following function:

1. function adminAllSafeTransfer()
2. function updateInsuranceRate()
3. function settlementOfClaims()

In the contract `VoucherNft`, the role `operator` has the authority over the following function:

1. function setNftUri()
2. function mintNft()

Any compromise to the accounts may allow a hacker to take advantage of this authority and causes security problems.

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.

Noted: Recommend considering the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

The team response: Some permissions have been cancelled, and the remaining permissions are due to business needs.

CSI-01 | Missing Validation

Category	Severity	Location	Status
Logical Issue	Major	CoinSalesIDO.sol: 418~427, 429~431	Resolved

Description

The `sellingToken` can be changed after `startTime`. So `sellingToken` can be changed after the user has made a purchase. The user may not get the `sellingToken` that he wanted to buy.

Recommendation

We recommend adding a check to ensure the `sellingToken` can only be changed before `startTime`.

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

CSI-02 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	CoinSalesIDO.sol: 264~265	Resolved

Description

The `userFloor` means the minimum amount of tokens one user can buy. The `userQuota` means the maximum amount of tokens one user can buy. So the `userFloor` should be less than `userQuota`.

Recommendation

Consider adding a check as below:

```
1     require(intArr[7] > intArr[8], "userQuota should be greater than userFloor!");
```

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

CSI-03 | Lack Of Reasonable Boundary

Category	Severity	Location	Status
Volatile Code	Minor	CoinSalesIDO.sol: 277, 372~416	Partially Resolved

Description

1. The variable `yield` means the percentage a second party can get from a `CoinSalesIDO`. Its current range of values is `yield<=100`. We think its range of values should be more reasonable.
2. If `_isLockWarehouse == 2`, the selling token may be locked for a certain period of time. This time should be within a reasonable range.

Recommendation

We recommend adding reasonable upper and lower boundaries to all the configuration variables.

Alleviation

1. The range of values changed to `0<yield<100`.
2. The final unlock time limit is 10 years after the current time.

Code was applied in commit `a38539d19c200c900888315ad35dcd9acf938470`.

CSI-04 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	CoinSalesIDO.sol: 317~335	Acknowledged

Description

According to the logic, `startTime` should be less than `endTime`, `endTime` should be less than `claimTime` and `closeTime`. So there need a check before updating the variables `startTime`, `endTime`, `claimTime` and `closeTime`.

Recommendation

Consider adding a check to ensure the variables mentioned above have a reasonable value.

Alleviation

The team response: Because of the needs of business needs, there is no need to solve them.

CSI-05 | LockWarehouse Can Be Added After User Investment

Category	Severity	Location	Status
Logical Issue	Minor	CoinSalesIDO.sol: 372~416	Acknowledged

Description

The `LockWarehouse` can be added after user investment, so users may not know the lockdown program when they participate exchange.

Recommendation

It's best to add `LockWarehouse` before users participate exchange.

Alleviation

The team response: Because of the needs of business needs, there is no need to solve them.

CSI-06 | Condition Can Be Simplified

Category	Severity	Location	Status
Volatile Code	Informational	CoinSalesIDO.sol: 481	Resolved

Description

In Line481, `_amount` <= `investment.add(_amount)`. So Line481 can be simplied.

Recommendation

Consider modifying as below:

```
1      require(investment.add(_amount) <= quota, "Err_46");
```

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

FID-01 | Incorrect Logic In Function `customProductionIDO`

Category	Severity	Location	Status
Logical Issue	Medium	FactoryIDO.sol: 403~435	Acknowledged

Description

function `foundCoinSalesIdo`:

```
    _addressArray[3] = msg.sender;
    _addressArray[4] = superAdmin;
    _addressArray[5] = insurance;
    _addressArray[6] = address(this);

    require(isInsuranceToken(_addressArray[1]), "Illegal token!");
    _intArray[9] = uint256(ERC20(_addressArray[1]).decimals());
```

function `customProductionIDO`:

```
    _addressArray[0] = msg.sender;
    _addressArray[1] = superAdmin;
    _addressArray[2] = insurance;
    _addressArray[3] = address(this);
    _addressArray[4] = _assemblyLine;

    require(isInsuranceToken(_addressArray[5]), "Illegal token!");
    _intArray[0] = uint256(ERC20(_addressArray[5]).decimals());
```

The functions `foundCoinSalesIdo` and `customProductionIDO` have similar features. So the parameters passed to `assemblyLine` should be the same. But the real parameters shown above are different. The parameters in `customProductionIDO` are incorrect. Please correct the parameters.

Additionally, The newly created `ido` is not added to the `firstPartyPoolArray` in function `customProductionIDO`, like in `foundCoinSalesIdo`.

Recommendation

We recommend to correct the logic in function `customProductionIDO`.

Alleviation

The team response: `customProductionIDO` is a later reserved extension function and does not correspond to the `productionIdo` function in `AssemblyLine`, it is a reserved interface for the new `AssemblyLine` later.

ITP-01 | Possible Loss Of Accuracy In Function `joinInsurance`

Category	Severity	Location	Status
Logical Issue	Medium	Insurance.sol: 169, 172	Resolved

Description

Variables `totalInsuredAmount`, `userInsuredTotal` are without precision. The default value of `insuranceRate` is 1%. When the investment amount is not an integer multiple of 100, `totalInsuredAmount` and `userInsuredTotal` will have a loss of precision.

For example: _investedAmount = 99 * 10 ** decimals; _amount = 0.99 * 10 ** decimals; _amount.div(10 ** decimals) = 0;

Recommendation

We recommend retaining the precision of `totalInsuredAmount` and `userInsuredTotal`.

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

ITP-02 | Incorrect Operator Used

Category	Severity	Location	Status
Logical Issue	Medium	Insurance.sol: 169, 172	Resolved

Description

```
1     totalInsuredAmount = totalInsuredAmount.add(_amount.div(10 * decimals));
2
3     userInsuredTotal[_user] = userInsuredTotal[_user].add(_amount.div(10 * decimals));
```

The operator in Line169 and Line172 should be **, but not *.

Recommendation

We recommend using the correct operator.

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

ITP-03 | Inadequate Validation

Category	Severity	Location	Status
Logical Issue	Medium	Insurance.sol: 150~179	Resolved

Description

The validation in `Insurance.joinInsurance` is not sufficient. Users can forge their own contracts into `CoinSalesIDO` to mint themselves NFT.

Recommendation

We recommend using of more adequate calibration `FactoryIDO.isVoucherIdo(msg.sender)` to ensure the `CoinSalesIDO` is a valid one.

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

TPC-01 | Permission Issues For Super Admin

Category	Severity	Location	Status
Logical Issue	Major	FactoryIDO.sol: 150~154 Insurance.sol: 111~116	Resolved

Description

The function `updateAdmin` can only be called by the `superAdmin`. The functions `removeAuth` and `addAuth` can only be called by the `owner`. But the `superAdmin` is set in `constructor`, maybe not the `owner`. So the `superAdmin` may not have the permission to call the functions `removeAuth` and `addAuth`.

Recommendation

We recommend assigning `owner` to `superAdmin` in constructor.

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

TPC-02 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	AssemblyLine.sol: 20, 25 CoinSalesIDO.sol: 420 FactoryIDO.sol: 99, 102~106, 152 Insurance.sol: 61~63	Acknowledged

Description

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

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:
For example:

```
1     require(_lineLeader != address(0), "lineLeader can not be zero address!");
```

Alleviation

The team response: No need to resolve.

TPC-03 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	AssemblyLine.sol: 2 token/VoucherNft.sol: 2 CoinSalesIDO.sol: 2 Insurance.sol: 2 FactoryIDO.sol: 2	Acknowledged

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 different compiler versions. 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;
```

Alleviation

The team response: No need to resolve.

TPC-04 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	AssemblyLine.sol: 23, 28 CoinSalesIDO.sol: 228, 300, 308, 313, 317, 337, 341, 345, 356, 372, 418, 429, 433, 438, 452, 515, 530 Insurance.sol: 88, 111, 118 FactoryIDO.sol: 142, 150, 176, 181, 215, 222, 252, 295	Acknowledged

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.

Alleviation

The team response: No need to resolve.

TPC-05 | Variables That Could Be Declared As Immutable

Category	Severity	Location	Status
Gas Optimization	Informational	AssemblyLine.sol: 11 CoinSalesIDO.sol: 27 Insurance.sol: 20, 22 FactoryIDO.sol: 19	Acknowledged

Description

The linked variables assigned in the constructor can be declared as `immutable`. Immutable state variables can be assigned during contract creation but will remain constant throughout the lifetime of a deployed contract. A big advantage of immutable variables is that reading them is significantly cheaper than reading from regular state variables since they will not be stored in storage.

Recommendation

We recommend declaring these variables as immutable. Please note that the `immutable` keyword only works in Solidity version `v0.6.5` and up.

Alleviation

The team response: No need to resolve.

VNT-01 | Missing Creation In Function `queryUserAllNft`

Category	Severity	Location	Status
Language Specific	Major	token/VoucherNft.sol: 74~79	Resolved

Description

In function `queryUserAllNft`, `nftArray` is not created. When this method be called there will be an error.

Recommendation

Consider modifying as below:

```
    nftArray = new NftInfo[](_ids.length);
    for (uint256 i = 0; i < _ids.length; i++) {
        nftArray[i] = nftInfo[_ids[i]];
    }
    return nftArray;
```

Alleviation

The development team has resolved this issue in commit `a38539d19c200c900888315ad35dcd9acf938470`.

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.

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.

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.