CERTIK

Preliminary Comments

Huh-governance

Mar 18th, 2022

Table of Contents

Summary

This report has been prepared for Huh-governance to discover issues and vulnerabilities in the source code of the Huh-governance 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	Huh-governance
Description	Huh-governance
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/HUH-Token/huh-governance/tree/main/contracts
Commit	ac95c54ca8249ab4ea6218d40a68deff7af0ed11

Audit Summary

Delivery Date	Mar 18, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

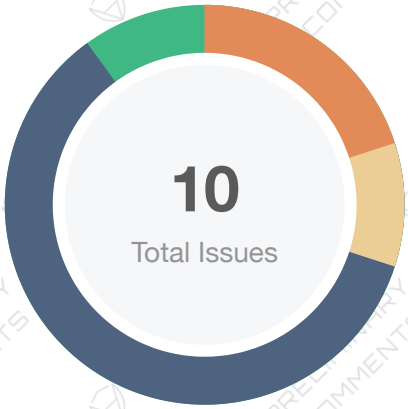
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	2	2	0	0	0	0	0
Medium	0	0	0	0	0	0	0
Minor	1	1	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
HUH	HUHGovernance.sol	4ab2815947135f5628a066b3b6004ee1ca450a452aad2a781c6ee2d6d4481a5e
HUG	HUHGovernance_V2.sol	614a279297f1ef1eb0194ddc9127d3d6405119ba0bd457def73cfeb83acdb2b6
IHU	Import.sol	e72de56a324ac6b157cb4c1bbc839094c68833146a45befa7fa80c5e633f4c38
THU	Timestamp.sol	a02725ea4f0d6a3a143e64a60c3390eccd7a5a7728d1062b84225f3748b7fd02
THH	Token.sol	be6feaead7897a8c211282582dde1bf17fd23f93e9150c8303a1f7185a64b69d
TTL	TokenTimeLock.sol	7cd37c01a91401e17ec7cd359ecd7ee52d2314877b8beb560b46b2858744b4b5

Findings



Critical	**0** (0.00%)
Major	**2** (20.00%)
Medium	**0** (0.00%)
Minor	**1** (10.00%)
Informational	**6** (60.00%)
Discussion	**1** (10.00%)

ID	Title	Category	Severity	Status
HUH-01	Redundant Code Components	Volatile Code	Informational	Pending
HUH-02	Improper Usage of `public` and `external` Type	Gas Optimization	Informational	Pending
HUH-03	Unlocked Compiler Version	Language Specific	Informational	Pending
HUH-04	Variables That Could Be Declared as Immutable	Gas Optimization	Informational	Pending
HUH-05	Centralization Related Risks	**Centralization / Privilege**	**Major**	Pending
THH-01	Non-standard ERC20 contract	Volatile Code	Major	Pending
THH-02	Variables That Could Be Declared as `constant`	Gas Optimization	Informational	Pending
THH-03	Incorrect ERC-20 Interface	Language Specific	Minor	Pending
THU-01	Discussion of the function `caculateYearsDeltatime()`	Mathematical Operations, Logical Issue	Discussion	Pending
TTL-01	Missing Emit Events	Coding Style	Informational	Pending

HUH-01 | Redundant Code Components

Category	Severity	Location	Status
Volatile Code	Informational	HUHGovernance.sol: 27 HUHGovernance_V2.sol: 27 Token.sol: 19	Pending

Description

The linked statements do not affect the functionality of the codebase and appear to be either leftovers from test code or older functionality.

Recommendation

We advise to remove the redundant statements for production environments.

HUH-02 | Improper Usage Of `public` And `external` Type

Category	Severity	Location	Status
Gas Optimization	Informational	Timestamp.sol: 7, 12 TokenTimeLock.sol: 50, 71, 89 HUHGovernance.sol: 47, 51, 55, 64, 68, 72, 88, 97, 101, 105 HUHGovernance_V2.sol: 37, 41, 45, 49, 53, 57, 61, 78, 87, 91, 95	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.

HUH-03 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	Timestamp.sol: 3 Import.sol: 2 ERC20Mock.sol: 3 TokenTimeLock.sol: 3 HUHGovernance.sol: 1 HUHGovernance_V2.sol: 1 Token.sol: 5	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 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;
```

HUH-04 | Variables That Could Be Declared As Immutable

Category	Severity	Location	Status
Gas Optimization	Informational	HUHGovernance_V2.sol: 25 Token.sol: 19	Pending

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.

HUH-05 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	HUHGovernance.sol HUHGovernance_V2.sol	Pending

Description

In the contract `HUHGovernance/HUHGovernance_V2`, the role `proxy` has authority over the following functions:

- `upgradeTo()/upgradeToAndCall()`: change the implementation of the contract

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

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

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

Short Term:

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

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

Long Term:

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

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

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

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

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.

THH-01 | Non-standard ERC20 Contract

Category	Severity	Location	Status
Volatile Code	Major	Token.sol	Pending

Description

ERC20 standard as defined in the EIP contains several functions that a compliant token must be able to implement.

- TotalSupply: provides information about the total token supply
- BalanceOf: provides account balance of the owner's account
- Transfer: executes transfers of a specified number of tokens to a specified address
- TransferFrom: executes transfers of a specified number of tokens from a specified address
- Approve: allow a spender to withdraw a set number of tokens from a specified account
- Allowance: returns a set number of tokens from a spender to the owner1

In the contract `Token`, the functions `transferFrom()`, `approve()` and `allowance()` are not implemented.

Recommendation

We recommend realizating ERC20 standard as defined in the EIP.

THH-02 | Variables That Could Be Declared As `constant`

Category	Severity	Location	Status
Gas Optimization	Informational	Token.sol: 12, 13, 16	Pending

Description

The linked variables could be declared as `constant` since these state variables are never modified.

Recommendation

We recommend to declare these variables as `constant`.

THH-03 | Incorrect ERC-20 Interface

Category	Severity	Location	Status
Language Specific	Minor	Token.sol: 42	Pending

Description

Incorrect return values for ERC-20 functions. A contract compiled with Solidity > 0.4.22 interacting with these functions will fail to execute them, as the return value is missing.

```
Token (Token.sol#9-62) has incorrect ERC20 function
interface:Token.transfer(address,uint256) (Token.sol#42-51)
```

File: contracts/Token.sol (Line 42, Contract `Token`)

```
    function transfer(address to, uint256 amount) external {
```

Recommendation

We recommend setting the appropriate return values and types for the defined ERC-20 functions.

THU-01 | Discussion Of The Function `caculateYearsDeltatime()`

Category	Severity	Location	Status
Mathematical Operations, Logical Issue	Discussion	Timestamp.sol: 12	Pending

Description

According to the following codes, the function `caculateYearsDeltatime()` is used to calculate the seconds of the given years.

```
12      function caculateYearsDeltatime(uint _years) public pure returns (uint256){
13          uint oneDay = 1 days;
14          return ((_years * 3652425 + 5000)/ 10000) * oneDay;
15      }
```

In solidity, the division truncation problem exists for division operations. The calculation `(_years * 3652425 + 5000)/ 10000` uses division truncation to calculate the number of days for the given years.

However, the result of 1st,2nd,4th year is 365 days, the result of 3rd year is 366 days, and so on. This does not correspond to the common calculation of the number of days for the given years.

Recommendation

We recommend stating for this.

TTL-01 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	TokenTimeLock.sol: 89~98	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.

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.

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.