Code Security Assessment

Anonverse-audit

Jan 17th, 2022

Table of Contents

Summary

This report has been prepared for Anonverse-audit to discover issues and vulnerabilities in the source code of the Anonverse-audit 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	Anonverse-audit
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/Anonymous-Game/anon-core/tree/master/contracts
Commit	368447dde109f0eb2d8ab63a2aef34964ac99a0c 7f3d289d11e969b26514200f7e64dda0dd2bf9b2

Audit Summary

Delivery Date	Jan 17, 2022
Audit Methodology	Static Analysis, Manual Review

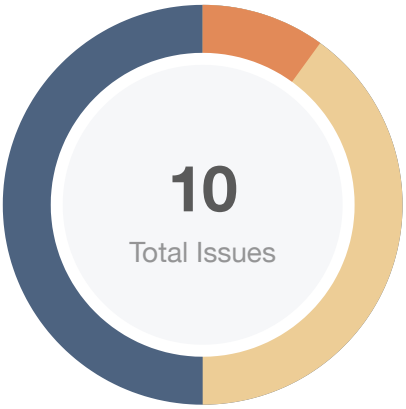
Vulnerability Summary

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

Audit Scope

ID	File	SHA256 Checksum
ATA	ANON/AnonverseToken.sol	51363272d80052247743f1ee2cdc9d7ac0e05fb3a418a020fd634cdd244d77da
DAC	donate/Donate.sol	6faf9e3247242b8c97bc1240e8b596434aaedae018e1c603df63d8ec11ab3dc9

Findings



Critical	**0** (0.00%)
Major	**1** (10.00%)
Medium	**0** (0.00%)
Minor	**4** (40.00%)
Informational	**5** (50.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
ACP-01	Unlocked Compiler Version	Language Specific	Informational	Acknowledged
ACP-02	Function Visibility Optimization	Gas Optimization	Informational	Acknowledged
ACP-03	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
DAC-01	Potential Front-Running Risk	Volatile Code	Minor	Acknowledged
DAC-02	Hardcode Address	Logical Issue	Informational	Acknowledged
DAC-03	Lack of Input Validation	Volatile Code	Minor	Resolved
DAC-04	Lack of Input Validation	Volatile Code	Minor	Resolved
DAC-05	Strict Conditional	Control Flow	Informational	Acknowledged
DAC-06	Missing Emit Events	Gas Optimization	Informational	Acknowledged
DAC-07	Incompatibility With Deflationary Tokens	Logical Issue	Minor	Resolved

ACP-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	ANON/AnonverseToken.sol: 2 donate/Donate.sol: 3	Acknowledged

Description

The contract has an 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 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

It is a general practice to instead lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and be able to identify ones more easily. We recommend locking the compiler at the lowest possible version that supports all the capabilities wished by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

No alleviation.

ACP-02 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	Informational	donate/Donate.sol: 46 ANON/AnonverseToken.sol: 9	Acknowledged

Description

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

Recommendation

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

Alleviation

No alleviation.

ACP-03 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	ANON/AnonverseToken.sol: 9 donate/Donate.sol: 69, 77, 90, 95, 99, 104	Acknowledged

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the considerations of the administrator team's anonymity.

In the contract `AnonverseToken`, the role `owner` has the responsibility to notify users about the following capabilities:

- mint uncapped tokens to anyone through `mint()`

In the contract `Donate`, the role `owner` has the responsibility to notify users about the following capabilities:

- set `feeReceiver` through `setFeeReceiver()`
- set `receiveToken` through `setReceiveToken()`
- transfer any token to anyone through `transferSourceToken()`
- transfer ETH to anyone through `transferETH()`
- change `startTime` through `changeStartTime()`
- change `endTime` through `changeEndTime()`

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.

Alleviation

`[Client]`: We will transfer owner rights to multisig wallet address of community when the contract will be deployed and stable.

DAC-01 | Potential Front-Running Risk

Category	Severity	Location	Status
Volatile Code	Minor	donate/Donate.sol: 46	Acknowledged

Description

Malicious hackers may observe the pending transaction which will execute the `initialize()` function and launch a similar transaction with the hacker's address, set `startTime`、`endTime`、`receiveToken`、`minDonatedAmount` and `feeReceiver` of the contract.

Recommendation

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

Alleviation

`[Client]`: We are using upgradeable contract framework of OpenZeppelin, which does not have this problem with its standard deployments.

DAC-02 | Hardcode Address

Category	Severity	Location	Status
Logical Issue	Informational	donate/Donate.sol: 61, 65	Acknowledged

Description

There are many hardcode addresses in this codebase.

Recommendation

We advise changing to the correct address before the contract is deployed onto blockchain.

Alleviation

No alleviation.

DAC-03 | Lack of Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	donate/Donate.sol: 104	Resolved

Description

According to the function `changeStartTime()`'s logic, the function `changeEndTime()` should check that `startTime` is less than `_endTime`.

Recommendation

We advise the client to add a validation for `_endTime`.

Alleviation

The client heeded our advice and resolved this issue in commit : 7f3d289d11e969b26514200f7e64dda0dd2bf9b2.

DAC-04 | Lack of Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	donate/Donate.sol: 99	Resolved

Description

According to the function `initialize()`'s logic, the function `changeStartTime()` should check that `block.timestamp` is less than `_startTime`.

Recommendation

We advise the client to add a validation for `_startTime`.

Alleviation

The client heeded our advice and resolved this issue in commit : 7f3d289d11e969b26514200f7e64dda0dd2bf9b2.

DAC-05 | Strict Conditional

Category	Severity	Location	Status
Control Flow	Informational	donate/Donate.sol: 52, 55	Acknowledged

Description

If `block.timestamp` is less than `1642251600`, calling `initialize()` will be revert.

Recommendation

We advise the client to check the accuracy of `startTime` and `endTime`.

Alleviation

No alleviation.

DAC-06 | Missing Emit Events

Category	Severity	Location	Status
Gas Optimization	Informational	donate/Donate.sol: 69, 77, 99, 104	Acknowledged

Description

Functions that affect the status of sensitive variables should be able to emit events as notifications to customers.

Recommendation

We advise the client to add events for sensitive actions, and emit them.

Alleviation

No alleviation.

DAC-07 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Logical Issue	Minor	donate/Donate.sol: 113	Resolved

Description

When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged transaction fee. As a result, an inconsistency in the amount will occur and the transaction may fail due to the validation checks.

Recommendation

We advise the client to regulate tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

The client heeded our advice and resolved this issue in commit : 7f3d289d11e969b26514200f7e64dda0dd2bf9b2.

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.

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.

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.