CERTIK

Security Assessment

stabledoc Token

Apr 8th, 2022

Table of Contents

Summary

This report has been prepared for stabledoc Token to discover issues and vulnerabilities in the source code of the stabledoc Token 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	stabledoc Token
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/address/0xaff33F2b4329e5aB0Fcb951A150373c332004e11 https://bscscan.com/address/0x159372cc202d2d29d349e608a1ae6daf6482c304
Commit	

Audit Summary

Delivery Date	Apr 08, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

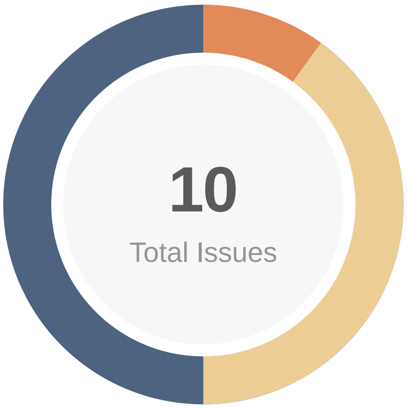
Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	1	0	0	1	0	0	0
● Medium	0	0	0	0	0	0	0
● Minor	4	0	0	2	0	0	2
● Informational	5	0	0	1	0	0	4
● Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
RGC	contracts/libs/utils/ReentrancyGuard.sol	bc91b68b4521a978bee8124368457df42c784f8bc851aa7a96b08195a8be85aa
SSM	contracts/libs/math/SignedSafeMath.sol	7446c74eb177831bbe10855d1eafe6d765f42d5b0bfa9c3d2542e0a002b9aa11
CGS	contracts/libs/GSN/Context.sol	8f72c714a7a1017f2c0aff7829297d3afe409b548adb14091379379fc3c5af28
TOK	contracts/libs/token	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
IBE	contracts/libs/token/BEP20/IBEP20.sol	64b105611fc126e5645069d30628943ce6b2a62fa2a7cfedcad16af60839660f
SMC	contracts/libs/math/SafeMath.sol	9427d3920994969cae7bc614b3f55893c6d70b2266a10974c97a8fd9241af0a7
BMC	contracts/libs/math/BoringMath.sol	0f9faff4a11d4e497f8df3c03e03534e81d128577fa19b39224bcca935855867
ACC	contracts/libs/access	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
LIB	contracts/libs	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
SBE	contracts/libs/token/BEP20/SafeBEP20.sol	6c36ba3150db9ff8aeadadba99654049dd437c1655cf109ed58b521018d0d3d4
IRC	contracts/libs/interfaces/IRewarder.sol	c7e08015091bb1588bccb8e6d3c753a5bc9fc86b3945f2cd3b492d035c3360b4
SSC	contracts/StabledocStaking.sol	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
INT	contracts/libs/interfaces	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
UTI	contracts/libs/utils	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
GSN	contracts/libs/GSN	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
ADD	contracts/libs/utils/Address.sol	1ab09ef3ee93a4090565345c9be9c8030b5772b9e37fe220c42828b80c5dd4c8

ID	File	SHA256 Checksum
OWN	contracts/libs/access/Ownable.sol	72a9fa3a6e71427774983f5d289e1ab88a8f5f014d12de4310a5252f73eaa813
MAT	contracts/libs/math	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb
BEP	contracts/libs/token/BEP20	7ea1acd93c81faed01ed4f7d6b644b951ad87ed1c8717a3be5b73a71076877eb

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
CON-01	Centralization Related Risks	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
CON-02	Function Visibility Optimization	Gas Optimization	● Informational	✓ Resolved
SSC-01	Recommended Explicit Pool Validity Checks	Logical Issue	● Minor	✓ Resolved
SSC-02	Incompatibility With Deflationary Tokens	Logical Issue	● Minor	ⓘ Acknowledged
SSC-03	Third Party Dependencies	Control Flow	● Minor	ⓘ Acknowledged
SSC-04	Lack of Zero Address Validation	Volatile Code	● Minor	✓ Resolved
SSC-05	Missing Emit Events	Coding Style	● Informational	✓ Resolved
SSC-06	Comparison to A Boolean Constant	Gas Optimization	● Informational	✓ Resolved
SSC-07	Redundant Variable Initialization	Coding Style	● Informational	✓ Resolved
SSC-08	Discussion For Function `onSdtReward()`	Volatile Code	● Informational	ⓘ Acknowledged

CON-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	contracts/libs/access/Ownable.sol (v1): 55, 64 contracts/StabledocStaking.sol (v1): 140, 148, 170, 406	ⓘ Acknowledged

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the consideration of the administrator team’s anonymity.
The `owner` of `StabledocStaking` has the responsibility to notify users about the following capabilities:

- set `emergencyWithdrawable` through `setEmergencyWithdrawable()`
- add a new LP to the pool through `add()`
- update the given pool's SDT allocation point and `IRewarder` contract through `set()`
- withdraw SDT reward through `withdrawSdtReward()`
- set `isBlackListed` through `setBlackListed()`

The `owner` of `Ownable` has the responsibility to notify users about the following capabilities:

- renounce ownership through `renounceOwnership()`
- transfer ownership through `transferOwnership()`

Any compromise to the privileged 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.

Alleviation

`[Client]`: As a team we have agreed to continue with single wallet signatory as against the multi signatories/time lock recommended.

CON-02 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	● Informational	contracts/StabledocStaking.sol (v1): 99, 140, 148, 170, 223, 256, 290 , 333, 356 contracts/libs/access/Ownable.sol (v1): 36, 55, 64	✓ Resolved

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

The client revised the code and resolved this issue in BscScan.

SSC-01 | Recommended Explicit Pool Validity Checks

Category	Severity	Location	Status
Logical Issue	● Minor	contracts/StabledocStaking.sol (v1): 170, 205, 223, 256, 290, 333, 356	✓ Resolved

Description

There's no sanity check to validate if a pool is existing.

Recommendation

We advise the client to recheck the function.

Alleviation

The client revised the code and resolved this issue in BscScan.

SSC-02 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Logical Issue	● Minor	contracts/StabledocStaking.sol (v1): 241	ⓘ Acknowledged

Description

When standard ERC20 deflationary tokens are transferred, the expended amount may be less than the received amount due to the transaction fee mechanism. As a result of such inconsistency, the depositing transaction will fail the validation checks in `safeTransferFrom()` and be reverted.

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

No alleviation.

SSC-03 | Third Party Dependencies

Category	Severity	Location	Status
Control Flow	● Minor	contracts/StabledocStaking.sol (v1): 238, 273, 316, 347, 374	ⓘ Acknowledged

Description

The contract is serving as the underlying entity to interact with third-party protocols. The scope of the audit would treat those 3rd party entities as black boxes and assume their functional correctness. However, in the real world, 3rd parties may be compromised and lead to assets being lost or stolen.

Recommendation

We encourage the team to constantly monitor the status of those 3rd parties to mitigate negative outcomes when unexpected activities are observed.

Alleviation

No alleviation.

SSC-04 | Lack Of Zero Address Validation

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/StabledocStaking.sol (v1): 223, 256, 290, 333, 356	✓ Resolved

Description

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

Recommendation

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

Alleviation

The client revised the code and resolved this issue in BscScan.

SSC-05 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	contracts/StabledocStaking.sol (v1): 140, 406	✓ Resolved

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

The client revised the code and resolved this issue in BscScan.

SSC-06 | Comparison To A Boolean Constant

Category	Severity	Location	Status
Gas Optimization	● Informational	contracts/StabledocStaking.sol (v1): 106	✓ Resolved

Description

A boolean is compared to a boolean constant while it can be used directly and does not need to be compared to true or false.

Recommendation

We advise removing the comparison to the boolean constant.

Alleviation

The client revised the code and resolved this issue in BscScan.

SSC-07 | Redundant Variable Initialization

Category	Severity	Location	Status
Coding Style	● Informational	contracts/StabledocStaking.sol (v1): 78	✓ Resolved

Description

All variable types within Solidity are initialized to their default `empty` value, which is usually they zeroed out representation.

Particularly:

- `uint` / `int`: All `uint` and `int` variable types are initialized at `0`
- `address`: All `address` types are initialized to `address(0)`
- `byte`: All `byte` types are initialized to their `byte(0)` representation
- `bool`: All `bool` types are initialized to `false`
- `ContractType`: All contract types (i.e. for a given `contract ERC20 {}` its contract type is `ERC20`) are initialized to their zeroed out address (i.e. for a given `contract ERC20 {}` its default value is `ERC20(address(0))`)
- `struct`: All `struct` types are initialized with all their members zeroed out according to this table

Recommendation

We advise that the linked initialization statements are removed from the codebase to increase legibility.

Alleviation

The client revised the code and resolved this issue in BscScan.

SSC-08 | Discussion For Function `onSdtReward()`

Category	Severity	Location	Status
Volatile Code	● Informational	contracts/StabledocStaking.sol (v1): 238	ⓘ Acknowledged

Description

The `user` parameter in the above code is passed the value of `to`, while the parameter passed in the other function in the contract is `msg.sender`.

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design.

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.

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.