Assure DeFi™

The Verification **Gold Standard ™**

Security Assessment

Rebirth Token

October 4, 2023

Audit Status: Pass

Audit Edition: Advance

ASSURE DEFI™

THE VERIFICATION GOLD STANDARD

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
Critical	Danger or Potential Problems.
High	Be Careful or Fail test.
Low	Pass, Not-Detected or Safe Item.
Informational	Function Detected

Manual Code Review Risk Results

Contract Privilege	Description
Buy Tax	0%
Sale Tax	0%
Cannot Sale	Pass
Cannot Sale	Pass
Max Tax	0%
Modify Tax	Pass
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Can Pause Trade?	Pass
Pause Transfer?	Not-Detected
Max Tx?	Pass
Is Anti Whale?	Not Detected
Is Anti Bot?	Not Detected

Contract Privilege	Description
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not Detected
Can Mint?	Pass
Is Proxy?	Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	Not Available
Self Destruct?	Not Detected
External Call?	Detected
Other?	Not Detected
Holders	1
Auditor Confidence	High Risk

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	Not Deployed
Name	Rebirth
Token Tracker	Rebirth (RBH)
Decimals	18
Supply	1,000,000,000
Platform	Ethereum
compiler	v0.8.21+commit.d9974bed
Contract Name	RebirthedToken
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://pastebin.com/aySe72m7
Payment Tx	Corporate

Main Contract Assessed Contract Name

Name	Contract	Live
Rebirth	Not Deployed	No

TestNet Contract Assessed Contract Name

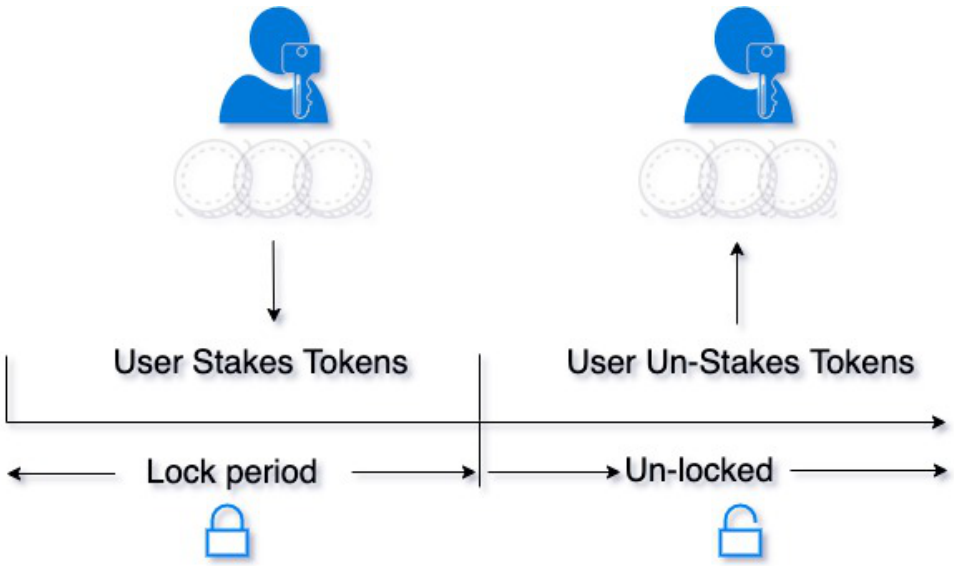
Name	Contract	Live
Rebirth	0x74D7fE0E79657C2C4C2920bAb1CAF743d1AE5d48	No

Solidity Code Provided

SolID	File Sha-1	FileName
Rebirth	c3f68e4bbd14b23bdbd26220a32dfc5765d86ae0	Rebirth.sol
Rebirth		
Rebirth		
Rebirth		

What is a Staking Contract

A smart contract which allows users to stake and un-stake a specified ERC20 token. Staked tokens are locked for a specific length of time (set by the contrat owner at the outset). Once the time period has elapsed, the user can remove their tokens again.



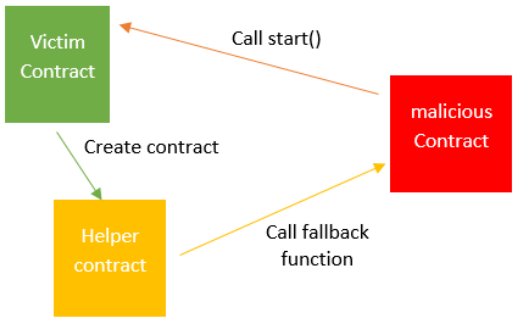
Reentrancy Check

The Project Owners of Rebirth have not configure the Reentrancy Guard library.

You can read more about Reentrancy issues in the following link.
Reentrancy After Istanbul.

We recommend the team to add the library to the contract to avoid potential issues.

We recommend the team to create a new contract with Reentrancy Guard added to the same.



Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	Rebirth.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	Rebirth.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	Rebirth.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	Rebirth.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	Rebirth.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	Rebirth.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	Rebirth.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	Rebirth.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	Rebirth.sol	L: 37 C: 22, L: 38 C: 23
SWC-109	Pass	Uninitialized Storage Pointer.	Rebirth.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	Rebirth.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	Rebirth.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	Rebirth.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	Rebirth.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-114	Pass	Transaction Order Dependence.	Rebirth.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	Rebirth.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	Rebirth.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	Rebirth.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	Rebirth.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	Rebirth.sol	L: 0 C: 0
SWC-120	Low	Potential use of block.number as source of randonmness.	Rebirth.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	Rebirth.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	Rebirth.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	Rebirth.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	Rebirth.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	Rebirth.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	Rebirth.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	Rebirth.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	Rebirth.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	Rebirth.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	Rebirth.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	Rebirth.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-132	Pass	Unexpected Ether balance.	Rebirth.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	Rebirth.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	Rebirth.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	Rebirth.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	Rebirth.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-108 - State Variable Default Visibility

CWE-710: Improper Adherence to Coding Standards

Description:

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

Remediation:

Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

References:

Ethereum Smart Contract Best Practices - Explicitly mark visibility in functions and state variables

Smart Contract Vulnerability Details

SWC-120 - Weak Sources of Randomness from Chain Attributes

CWE-330: Use of Insufficiently Random Values

Description:

Solidity allows for ambiguous naming of state variables when inheritance is used. Contract A with a variable x could inherit contract B that also has a state variable x defined. This would result in two separate versions of x, one of them being accessed from contract A and the other one from contract B. In more complex contract systems this condition could go unnoticed and subsequently lead to security issues.

Shadowing state variables can also occur within a single contract when there are multiple definitions on the contract and function level.

Remediation:

Using commitment scheme, e.g. RANDAO. Using external sources of randomness via oracles, e.g. Oraclize. Note that this approach requires trusting in oracle, thus it may be reasonable to use multiple oracles. Using Bitcoin block hashes, as they are more expensive to mine.

References:

How can I securely generate a random number in my smart contract?)

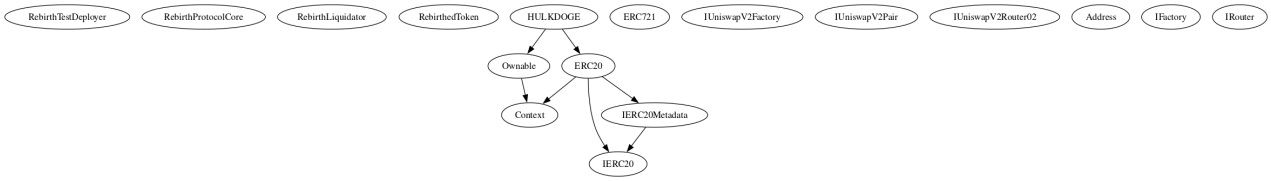
When can BLOCKHASH be safely used for a random number? When would it be unsafe?

The Run smart contract.

Inheritance

The contract for Rebirth has the following inheritance structure.

The Project has a Total Supply of 1,000,000,000



Smart Contract Advance Checks

ID	Severity	Name	Result	Status
RBH-01	Low	Potential Sandwich Attacks.	Pass	Not Detected
RBH-02	Informational	Function Visibility Optimization	Fail	Detected
RBH-03	Low	Lack of Input Validation.	Fail	Detected
RBH-04	High	Centralized Risk In addLiquidity.	Pass	Not Detected
RBH-05	Low	Missing Event Emission.	Fail	Detected
RBH-06	Low	Conformance with Solidity Naming Conventions.	Fail	Detected
RBH-07	Low	State Variables could be Declared Constant.	Pass	Not Detected
RBH-08	Low	Dead Code Elimination.	Pass	Not Detected
RBH-09	High	Third Party Dependencies.	Pass	Not Detected
RBH-10	High	Initial Token Distribution.	Pass	Not Detected
RBH-11	Critical	Contract use a custom Ownership library.	Pass	Remediated
RBH-12	High	Centralization Risks In The X Role	Pass	Not Detected
RBH-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
RBH-14	High	Unnecessary Use Of SafeMath	Pass	Not Detected
RBH-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
RBH-16	Medium	Taxes can be up to 100%	Pass	Not Detected
RBH-17	Informational	Conformance to numeric notation best practice.	Pass	Not Detected
RBH-18	Critical	Stop Transactions by using Enable Trade.	Pass	Not Detected

RBH-02 | Function Visibility Optimization.

Category	Severity	Location	Status
Gas Optimization	Informational	Rebirth.sol: L: 187 C: 24, L: 204 C: 23,L: 358 C: 14,L: 364 C: 14,L: 372 C: 14,L: 376 C: 14,L: 376 C: 14,L: 141 C: 14,L: 142 C: 14	Detected

Description

The following functions are declared as public and are not invoked in any of the contracts contained within the projects scope:

Function Name	Parameters	Visibility
CreatePool	address TokenAddress, address Pair Address, uint256 HoursTillOpen, uint256 LengthInHOurs, uint256 SoftCap, string memory TokenName, string memory TokenSymbol	Public
ClosePool	uint256 PoolID	Public
setSuperAdmin	address _newAdmin	Public
AddRemoveAdmin	address _newAdmin, bool AddRemove	Public
SetFreemintContract	address _FreemintContract	Public
SetLiquidator	address _Liquidator	Public
WithdrawRB		Public
UniswapRouter		Internal
UniswapFactory		Internal

The functions that are never called internally within the contract should have external visibility

Remediation

We advise that the function's 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.

References:

external vs public best practices.

RBH-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Low	Rebirth.sol: L: 187 C: 24, L: 265 C: 15, L: 269 C: 15, L: 273 C: 14, L: 277 C: 14, L: 282 C: 14,L: 288 C: 14, L: 292 C: 14,L: 296 C: 14, L: 309 C: 14	Detected

Description

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

The given input is missing the check for the .

Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
 require(receiver != address(0), "Receiver is the zero address");
...
...
require(value X limitation, "Your not able to do this function");
...
```

We also recommend customer to review the following function that is missing a required validation. .

RBH-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	Low	Rebirth.sol: L: 187 C: 24, L:204 C: 23, L: 265 C: 15, L: 269 C: 15, L: 273 C: 14, L: 277 C: 14, L: 282 C: 14, L: 288 C: 14, L: 292 C: 14,L: 296 C: 14, L: 301 C: 14, L: 309 C: 14	Detected

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

RBH-06 | Conformance with Solidity Naming Conventions.

Category	Severity	Location	Status
Coding Style	Low	Rebirth.sol: L: 402 C: 14,L: 389 C: 14,L: 601 C: 14,L: 610 C: 14	Detected

Description

Solidity defines a naming convention that should be followed. Rule exceptions: Allow constant variable name/symbol/decimals to be lowercase. Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
Freeminted
AddFreemint
Mint
Burn
```

Remediation

Follow the Solidity naming convention.

https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-convention

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	0	0	1
High	0	0	1
Medium	0	0	0
Low	3	0	0
Informational	1	0	0
Total	4	4	2

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/RebirthEth	Pass
Other	mail@rebirthprotocol.io	Pass
Website	https://rebirthprotocol.io/	Pass
Telegram	https://t.me/rebirthprotocoleth	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Assessment Results

Score Results

Review	Score
Overall Score	80/100
Auditor Score	80/100
Review by Section	**Score**
Manual Scan Score	17/23
SWC Scan Score	33 /37
Advance Check Score	30 /40

The Following Score System Has been Added to this page to help understand the value of the audit, the maximun score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 80 Points, if a project does not attain 80% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- The following code represents 3 contracts.ı
- https://goerli.etherscan.io/address/0x055502ea388b58cad59b94bdf7b0fda9f28c4b94#code RebirthLiquidatorı
- https://goerli.etherscan.io/address/0x923c8b791191e076e6f6fc1638cce2c1e8365789#code RebirthProtocolCoreı
- https://goerli.etherscan.io/address/0x74d7fe0e79657c2c4c2920bab1caf743d1ae5d48#code RebirthedTokenı
- The Customer improved the ownership from ownable library to use RBAC library(Access Control) which is a more secure library.ı
- We recommend the implementation of reentrancy to avoid potential issues with public functions.

Auditor Score =80
Audit Passed

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM 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 howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing 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 mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access 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 with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.