CERTIK

Security Assessment

RetroDoge

May 27th, 2022

Table of Contents

Summary

This report has been prepared for RetroDoge to discover issues and vulnerabilities in the source code of the RetroDoge 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	RetroDoge
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/address/0x335F07aC2c4363a5951427cbda38B0075455b2b5#code
Commit	

Audit Summary

Delivery Date	May 27, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

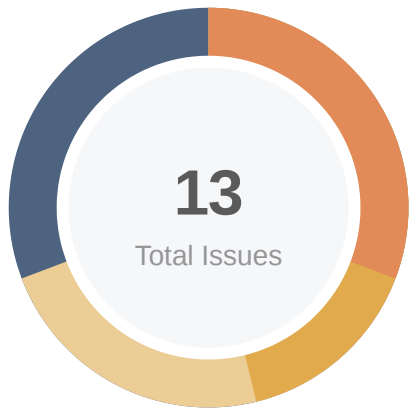
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	4	0	0	2	1	0	1
Medium	2	0	0	2	0	0	0
Minor	3	0	0	3	0	0	0
Informational	4	0	0	4	0	0	0
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
RDC	contracts/RetroDoge.sol	fec350c0369dde0aeb1b103677d9bbebd353cc0e1cf653f1b14a5e0a2310551a

Findings



Critical	**0** (0.00%)
Major	**4** (30.77%)
Medium	**2** (15.38%)
Minor	**3** (23.08%)
Informational	**4** (30.77%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Finance Model	Logical Issue	Medium	Acknowledged
RDC-01	Centralization Risks In RetroDoge.sol	**Centralization / Privilege**	**Major**	Mitigated
RDC-02	Initial Token Distribution	**Centralization / Privilege**	**Major**	Acknowledged
RDC-03	Centralized Risk In `addLiquidity`	**Centralization / Privilege**	**Major**	Acknowledged
RDC-04	Owner Can Pause Transfer	**Centralization / Privilege**	**Major**	Resolved
RDC-05	Update `launchMode` Mistakenly	Logical Issue	Medium	Acknowledged
RDC-06	Missing Zero Address Validation	Volatile Code	Minor	Acknowledged
RDC-07	Insufficient Tax Restriction	Logical Issue	Minor	Acknowledged
RDC-08	Potential Sandwich Attacks	Logical Issue	Minor	Acknowledged
RDC-09	Missing Emit Events	Coding Style	Informational	Acknowledged
RDC-10	State Variable Should Be Declared Constant	Gas Optimization	Informational	Acknowledged
RDC-11	Unused Event	Gas Optimization	Informational	Acknowledged
RDC-12	Related States Not Updated	Volatile Code	Informational	Acknowledged

GLOBAL-01 | Finance Model

Category	Severity	Location	Status
Logical Issue	Medium		Acknowledged

Description

The `RetroDoge` Protocol is a decentralized finance(Defi) token deployed on the BSC.

Most transactions will be charged fees including burnFee/stakeFee/marketingFee/liquidityFee. If at least one of sender and recipient is exclude from fee, the fees will be removed. The buy fee and sell fee can be set differently. The wallet transfer transaction is free. If the contract is in `launchMode`, every sell transaction will be charged `launchModeFees`. All the fee rates can be changed by owner at any time.

- initial totalBuyFees: 7%
- initial totalSellFees: 11%
- initial launchModeFees: 30%

The burnFee is burnt to dead wallet. The stakeFee is for `_stakingWalletAddress` in form of `RTDOGE` token. The marketingFee is for `_marketingWalletAddress` after converted to `MARKETING_TOKEN`. The liquidityFee is used to add liquidity. The LP token is for contract address, but the owner can withdraw it.

Recommendation

We recommend the team to be transparent to publish this feature to the community.

Alleviation

From the client:

This issue response must also be linked to RDC-05 as it affects the sell taxation currently in place for the contract, and RDC-07 as it affects the maximum taxation limitations coding.

For the sale fees, in general, we have openly notified our existing community before and during migration that the taxation would be changing and that there would be a high launch sales taxes, and our strategy behind such applications. A couple of examples are https://t.me/c/1635628013/13973, https://t.me/c/1635628013/8394. We have also notified our holders about the changeable taxes, and our strategy beyond that as well.

In acknowledging the mistake in RDC-05, we have identified that we cannot physically turn launch mode off, which then means the initial totalSellFees become redundant.

It also means that any restrictions we did place as per RDC-07 can be bypassed, as there is a set launchMode fees feature in this contract.

Because of this issue, we have corrected our taxation to 11% now, and the solution to be implemented in RDC-01 will also reduce the risk of malicious taxation changes.

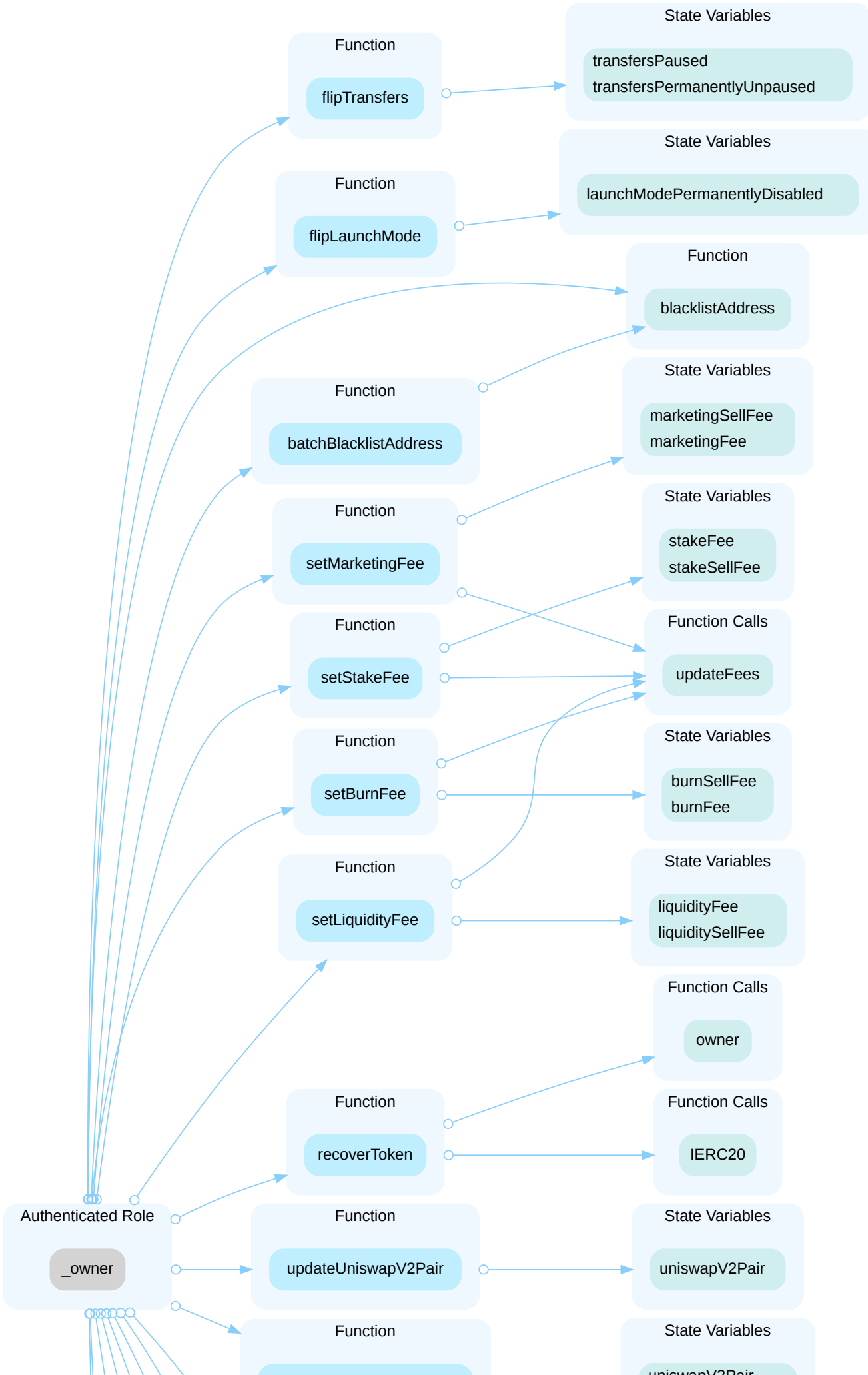
In the event of a relaunch or re-use of this contract, we will revise these issues and make the necessary adjustments to correct error RDC-07 and reduce maximum individual sell tax fees to 8% each.

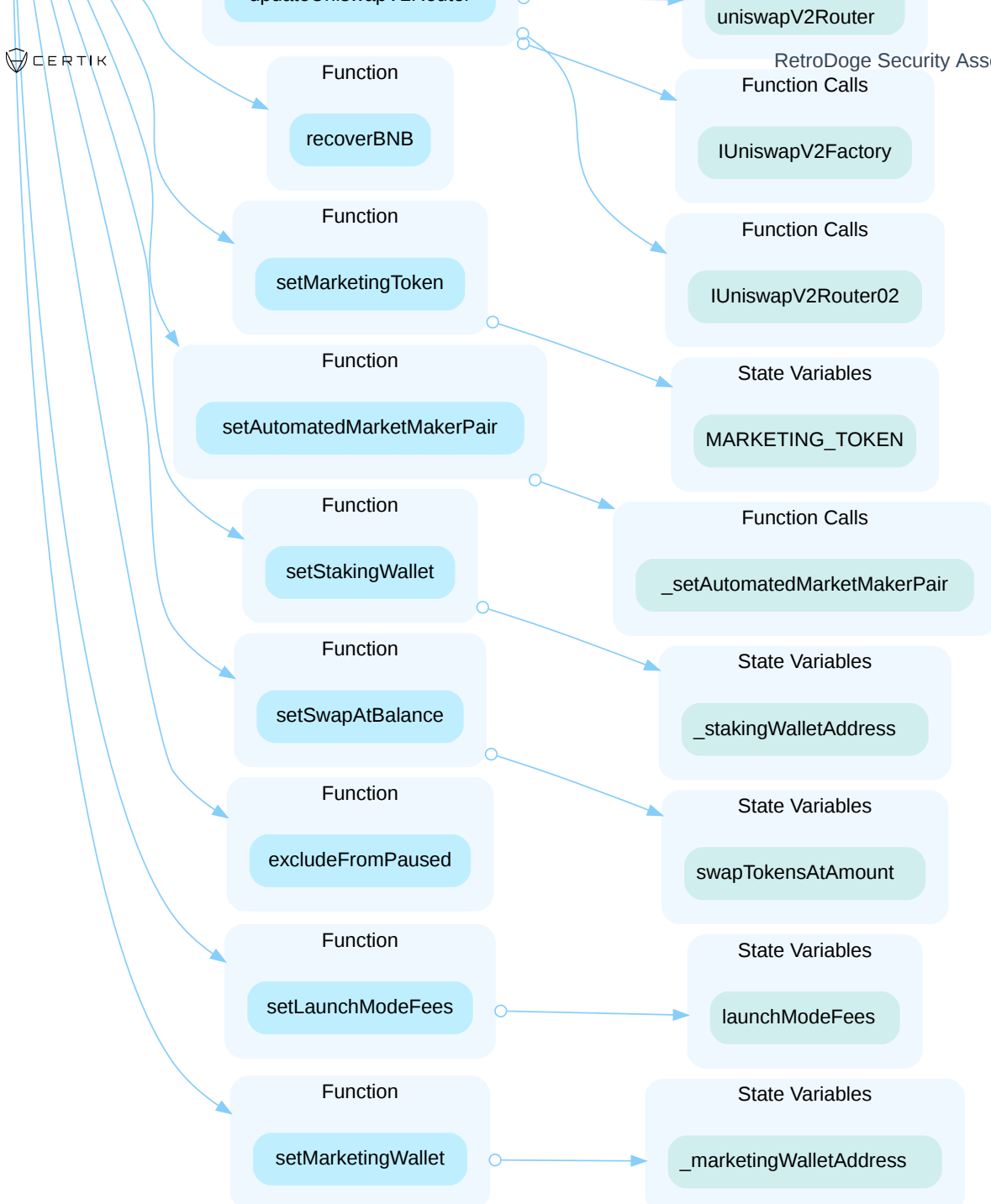
RDC-01 | Centralization Risks In RetroDoge.sol

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/RetroDoge.sol: 102, 110, 114, 118, 122, 132, 142, 152, 170, 174 , 178, 190, 196, 200, 206, 375, 380, 384, 388	Mitigated

Description

In the contract `RetroDoge` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.


State Variables
transfersPaused
transfersPermanentlyUnpaused
Function
flipTransfers
State Variables
launchModePermanentlyDisabled
Function
flipLaunchMode
Function
blacklistAddress
State Variables
marketingSellFee
marketingFee
Function
batchBlacklistAddress
State Variables
stakeFee
stakeSellFee
Function
setMarketingFee
Function Calls
updateFees
Function
setStakeFee
State Variables
burnSellFee
burnFee
Function
setBurnFee
State Variables
liquidityFee
liquiditySellFee
Function
setLiquidityFee
Function Calls
owner
Function
recoverToken
Function Calls
IERC20
Authenticated Role
_owner
Function
updateUniswapV2Pair
State Variables
uniswapV2Pair
Function
State Variables



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., multisignature 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

From the client:
We had noted this risk in our pre-migration strategic planning, and one of our risk mitigation strategies that we intend to implement is the short term option using Openzeppelin defender to manage it.

The reason it was not implemented before contract launch was because some of the functions needed immediate response capability. Empowering us to remove a ring of wallets that the developer of the last project had purchased and was using to artificially manipulate the token, claiming profits along the way. He may deny involvement in them, but the seeding patterns and his social media profiles before he deleted them told a different story.

We have implemented a time lock plus multi–signature combination for reducing the hazard of malicious contract manipulation by sending the contract to the time lock, then locking the multi signature so each

contract change will be delayed 50 hours.

Timelock: 0xFa8978d19e8a97d3A0065D7c61fA9Bbe9c70f231

Multi sig: 0xe223687f07B2373879Da54cCe0Cb98e516790C1c

We have also incorporated a second time lock to bypass the multi-signature wallet using the original deployer wallet in the event of an emergency, where the multi-signature is compromised and can no longer be able to be used (team member disappearance, death or keys to wallets lost).

This second time lock will have a 30 day restriction that also cannot be altered. The timelock is only operated by wallet address: 0x45Cde95A2a4e5c19E27fC5977E56b1e1479aBdec

Timelock: 0x96958e19F492Be9442980c827AD69761a1418EC1

This time lock plus multisig plus emergency measures system will be announced to the community in the spirit of transparency and mentioned in our to be released whitepaper.

Certik: Centralization Mitigation Info

- Time-lock(1)

This time lock is the owner of RetroDoge contract.

Time lock contract address:
https://bscscan.com/address/0xFa8978d19e8a97d3A0065D7c61fA9Bbe9c70f231

Time lock owner transfer transaction hash:
https://bscscan.com/tx/0xcae9207d124a5bae94de9d9db3bd98dfada991cf9ad60b401e642057019a11e3

- Multi-signature

This Multi-sign wallet is the PROPOSER_ROLE and EXECUTOR_ROLE of the Time-lock(1).

Multi-sign proxy address:
https://bscscan.com/address/0xe223687f07B2373879Da54cCe0Cb98e516790C1c

Internal multi-signature address:
bnb:0x3501745b30BB9d32a69e0990A7b977C5Aa56f758
bnb:0x45Cde95A2a4e5c19E27fC5977E56b1e1479aBdec
bnb:0x5f2413360772f237879f7194D40A8Ae060351758
bnb:0x9848EdA80717367a2820681Bb02C4Ae5902ee30B

- Time-lock(2)

This time lock is the PROPOSER_ROLE and EXECUTOR_ROLE of Time-lock(1) and is only operated by wallet address: 0x45Cde95A2a4e5c19E27fC5977E56b1e1479aBdec

Time lock contract address:
https://bscscan.com/address/0x96958e19F492Be9442980c827AD69761a1418EC1

RDC-02 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/RetroDoge.sol: 93	Acknowledged

Description

All of the `RTDOGE` tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute RTDOGE tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Alleviation

From the client:

It was noted that the new contract would mean centralized control of the supply for migration and as such we have been fairly clear in regards to how the tokens would be distributed or locked during the migration.

We also have locked up the LP and large amounts of the recovered supply of tokens from the ring of ex developer wallets via team finance.

https://www.team.finance/view-coin/0x335F07aC2c4363a5951427cbda38B0075455b2b5?name=RetroDoge&symbol=RTDOGE

Further usage of those tokens will be discussed with the community as the locks release, some have already been confirmed for use in the project, and not all of them will be unlocked at any one time to give investors comfort knowing we do not hold between 15 and 20% of remaining supply on one instance.

RDC-03 | Centralized Risk In `addLiquidity`

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/RetroDoge.sol: 369~377	ⓘ Acknowledged

Description

The `addLiquidity()` function calls the `uniswapV2Router.addLiquidityETH()` function with the `to` address specified as `address(this)` for acquiring the generated LP tokens from the `RTDOGE-WBNB` pool. As a result, over time the contract address will accumulate a significant portion of LP tokens.

However, the owner can withdraw all the LP tokens by function `recoverToken()`. If the `_owner` is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation

We strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

Indicatively, here are some feasible solutions that would also mitigate the potential risk:

- Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;
- Introduction of a DAO / governance / voting module to increase transparency and user involvement.

Alleviation

From the client:

We understand that the contract may collect tokens in that manner, and to allow the contract to be flushed and liquidity tokens added to the locker, the recover token call was implemented.

We will be following through with the risk mitigation strategy as set in RDC-01, to restrict access to these functions without a team agreement and a time lock period.

RDC-04 | Owner Can Pause Transfer

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/RetroDoge.sol: 192	Resolved

Description

The owner is able to call function `flipTransfers()` to pause all the transfer transactions.

Recommendation

We advise the client to carefully manage the `owner` account's private key and avoid any potential risks of being hacked. We also advise the client to adopt Multisig, Timelock, and/or DAO in the project to manage this specific account in this case.

Alleviation

The client has already set `transfersPermanentlyUnpaused` as true so that pause/unpause can not be switched now. Transaction hash as follow:
https://bscscan.com/tx/0x243b1a2be43e816053e2c793f46cf201bba8979ddf56ce47d9ef9d56473e62ca

From the client:

Fortunately, this is a non-issue as this pause/unpause function was pre-coded with a one-time use to launch the contract in a paused state of trade for the migration, enable liquidity adding and lockup, distribute excess tokens between project wallet addresses as explained to our community, and then unpause trade to start the new project.

We cannot stop trade now using this method.

RDC-05 | Update `launchMode` Mistakenly

Category	Severity	Location	Status
Logical Issue	Medium	contracts/RetroDoge.sol: 202	Acknowledged

Description

The `launchMode` can not be turned off because the `launchMode != launchMode` does not really switch the `launchMode`. This makes the `totalSellFees` useless.

Recommendation

We recommend to change as `launchMode = !launchMode`.

Alleviation

From the client:

We have addressed this issue by acknowledging that the coding cannot be flipped as was originally intended in GLOBAL-01.

Because we cannot change the launch mode, we have changed our launch mode fees to our intended value.

Because of that, this raises the RDC-07 concern, and to mitigate that risk, I refer back to the RDC-01 risk mitigation strategy.

In the event of a relaunch or re-use of this contract, we will revise these issues and make the necessary adjustments to correct them and ensure a smoother transition to normal trade.

RDC-06 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	contracts/RetroDoge.sol: 111, 115, 119, 381, 385	Acknowledged

Description

Addresses should be checked before assignment or external call to make sure they are not zero addresses.

File: contracts/RetroDoge.sol (Line 111, Function `RetroDoge.updateUniswapV2Pair`)

```
        uniswapV2Pair = newAddress;
```

- `newAddress` is not zero-checked before being used.

File: contracts/RetroDoge.sol (Line 115, Function `RetroDoge.setMarketingWallet`)

```
        _marketingWalletAddress = wallet;
```

- `wallet` is not zero-checked before being used.

File: contracts/RetroDoge.sol (Line 119, Function `RetroDoge.setStakingWallet`)

```
        _stakingWalletAddress = wallet;
```

- `wallet` is not zero-checked before being used.

File: contracts/RetroDoge.sol (Line 381, Function `RetroDoge.recoverBNB`)

```
        to.transfer(amount);
```

- `to` is not zero-checked before being used.

File: contracts/RetroDoge.sol (Line 385, Function `RetroDoge.setMarketingToken`)

```
        MARKETING_TOKEN = newToken;
```

- `newToken` is not zero-checked before being used.

Recommendation

We advise adding a zero-check for the passed-in address value to prevent unexpected errors.

Alleviation

No alleviation.

RDC-07 | Insufficient Tax Restriction

Category	Severity	Location	Status
Logical Issue	Minor	contracts/RetroDoge.sol: 73	Acknowledged

Description

It is not enough only to set the restriction that each fee can not be over 25%. If each fee value is set as `2499`, it will make the total fee almost close to 100%.

Recommendation

We recommend to add a restriction in function `updateFees` as well to ensure total fee no more than a proper value, such as 10%.

Alleviation

From the client:

With this deployment, the issue identified in RDC-05 becomes a higher tier problem for us due to the launch mode taxation arrangements being freely changeable.

Our risk mitigation strategy to address the taxation setting risk is set out in RDC-01 and should apply to this topic as well, as time locking and multi-signing new taxation values will prevent single point of failure hazards and unauthorized contract manipulation.

In the event of reuse or migration, we shall implement controls as set out in GLOBAL-01 to restore and strengthen the contract’s maximum taxation limits for even more investor security.

RDC-08 | Potential Sandwich Attacks

Category	Severity	Location	Status
Logical Issue	Minor	contracts/RetroDoge.sol: 315, 351, 367~368	Acknowledged

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by backrunning (after the transaction being attacked) a transaction to sell the asset.

The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- `swapTokensForEth()`
- `swapTokensForTokens()`
- `addLiquidity()`

Recommendation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

Alleviation

No alleviation.

RDC-09 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	contracts/RetroDoge.sol: 110, 114, 118, 122, 132, 142, 152, 170, 174, 178, 190, 196, 200, 206, 375, 380, 384, 388	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

No alleviation.

RDC-10 | State Variable Should Be Declared Constant

Category	Severity	Location	Status
Gas Optimization	Informational	contracts/RetroDoge.sol: 28, 51	Acknowledged

Description

State variables that never change should be declared as `constant` to save gas.

Recommendation

We recommend adding the `constant` attribute to state variables that never change.

Alleviation

No alleviation.

RDC-11 | Unused Event

Category	Severity	Location	Status
Gas Optimization	Informational	contracts/RetroDoge.sol: 59~60, 62, 70	Acknowledged

Description

The linked events are declared but never used.

Recommendation

We recommend to remove unused events or use them at proper places.

Alleviation

No alleviation.

RDC-12 | Related States Not Updated

Category	Severity	Location	Status
Volatile Code	Informational	contracts/RetroDoge.sol: 102~112, 114~118	Acknowledged

Description

After router and pair is updated in function `updateUniswapV2Router()` and `updateUniswapV2Pair()`, the function `_setAutomatedMarketMakerPair()` is not called to update the related states.

After wallet address is updated in function `setMarketingWallet()` and `setStakingWallet()`, the function `excludedFromPausedTransfersAndFees` is not called to update the related states.

Recommendation

We recommend to update the related states when updating sensitive variables.

Alleviation

No alleviation.

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.

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.