Code Security Assessment

Milk and Butter

Feb 2nd, 2022

Table of Contents

Summary

This report has been prepared for Milk and Butter to discover issues and vulnerabilities in the source code of the Milk and Butter 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	Milk and Butter
Platform	other
Language	Solidity
Codebase	https://github.com/cclaypool1/MilkToken/blob/main/MilkAndButter.sol
Commit	

Audit Summary

Delivery Date	Feb 02, 2022
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	1	0	0	1	0	0	0
Medium	1	0	0	1	0	0	0
Minor	2	0	0	2	0	0	0
Informational	7	0	0	7	0	0	0
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
MAB	projects/MilkToken/contracts/MilkAndButter.sol	0c4f0f84b1ff658a18a2be67909afcdd49242dfc75b6fe2956e04c605e912769

Findings

Critical	**0** (0.00%)
Major	**1** (9.09%)
Medium	**1** (9.09%)
Minor	**2** (18.18%)
Informational	**7** (63.64%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Fee Distribution	Volatile Code	Informational	Acknowledged
GLOBAL-02	Third Party Dependencies	Volatile Code	Minor	Acknowledged
MAB-01	Centralization Risk in MilkAndButter.sol	**Centralization / Privilege**	**Major**	Acknowledged
MAB-02	Variables that could be declared as `constant`	Gas Optimization	Informational	Acknowledged
MAB-03	Missing emit events	Coding Style	Informational	Acknowledged
MAB-04	Improper usage of `public` and `external` type	Gas Optimization	Informational	Acknowledged
MAB-05	Typos in the contract	Coding Style	Informational	Acknowledged
MAB-06	Incorrect Error Message	Logical Issue	Minor	Acknowledged
MAB-07	Contract gains non-withdrawable BNB via the `swapAndLiquify` function	Logical Issue	Medium	Acknowledged
MAB-08	Return Value Not Handled	Volatile Code	Informational	Acknowledged
MAB-09	The purpose of function `deliver`	Control Flow	Informational	Acknowledged

GLOBAL-01 | Fee Distribution

Category	Severity	Location	Status
Volatile Code	Informational	Global	Acknowledged

Description

There’re four types of fees depending on the situation: `_taxFee`, `_liquidityFee`, `_expenseFee` and `_charityFee`. The above fee rates are set to 1%, 1%, 5% and 3% when the contract is deployed, but they can be changed later. The `_expenseFee` and `_charityFee` are sent to corresponding wallets 0xbd05D7670611fd82ac0dB90BF48A5f01cF3B496F and 0xB30186581D1922a1A86E710161B3234c92945156. The generated LP tokens from the `MB-BNB` pool are separated to three portions. 10% of them are burned, 15% of them are sent to the manager(0x006F33031b14587bE5c6074006BA42E87756a282) and 75% of them are sent to a `lockedLiquidity` address(0x006F33031b14587bE5c6074006BA42E87756a282). All aforementioned addresses can be changed later.

Recommendation

This is the business logic of the MilkAndButter protocol, however, users should be aware of the fee distribution.

GLOBAL-02 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	Minor	Global	Acknowledged

Description

The contract is serving as the underlying entity to interact with third party PancakeSwap protocols. The scope of the audit treats 3rd party entities as black boxes and assume their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties can possibly create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Recommendation

We understand that the business logic of MilkAndButter requires interaction with PancakeSwap. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

[Milk and Butter Team]: We acknowledge that we are reliant on 3rd party protocols to handle certain aspects of how our token contract works. As such, we will continuously monitor the security of those protocols that we interact with.

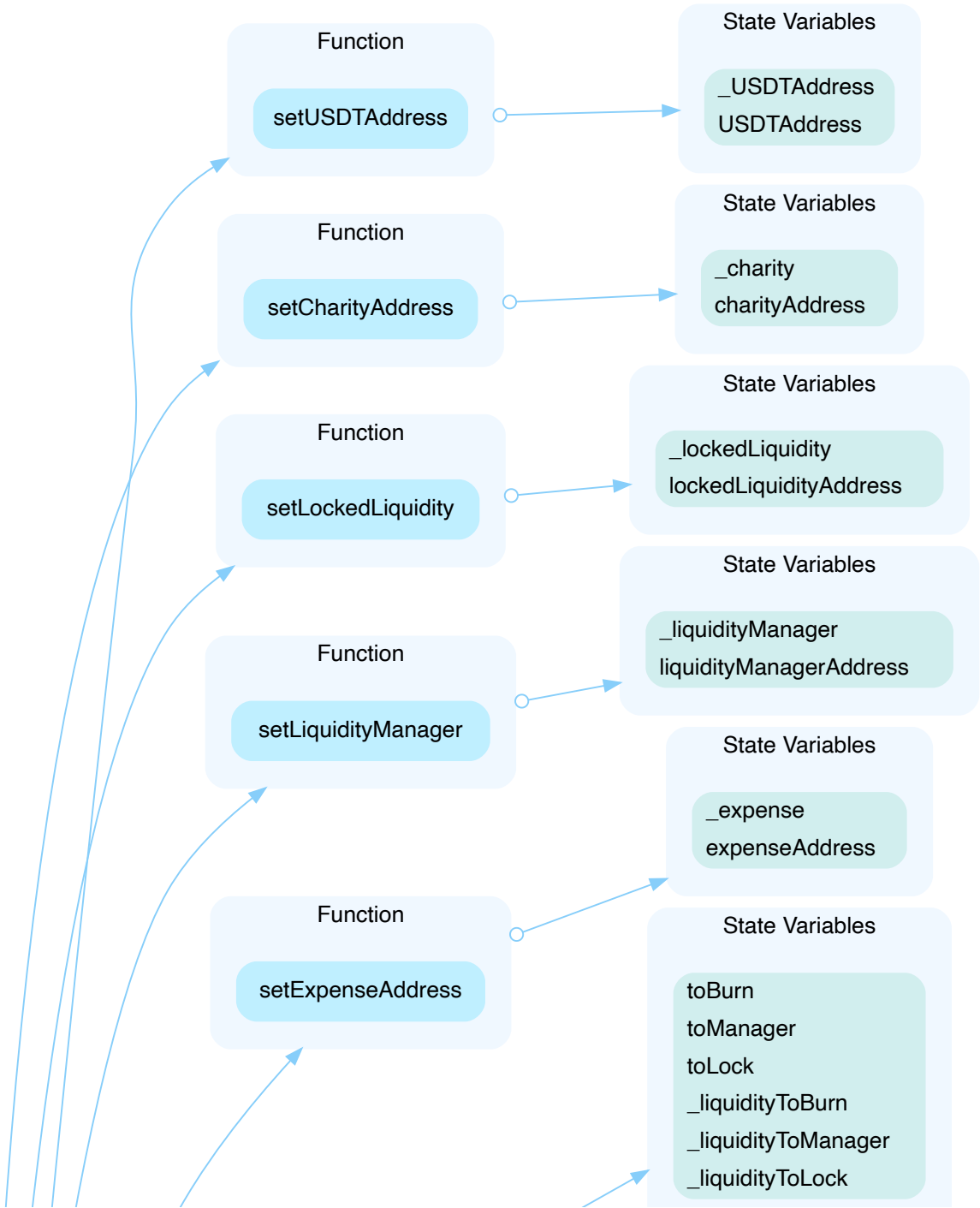
MAB-01 | Centralization Risk In MilkAndButter.sol

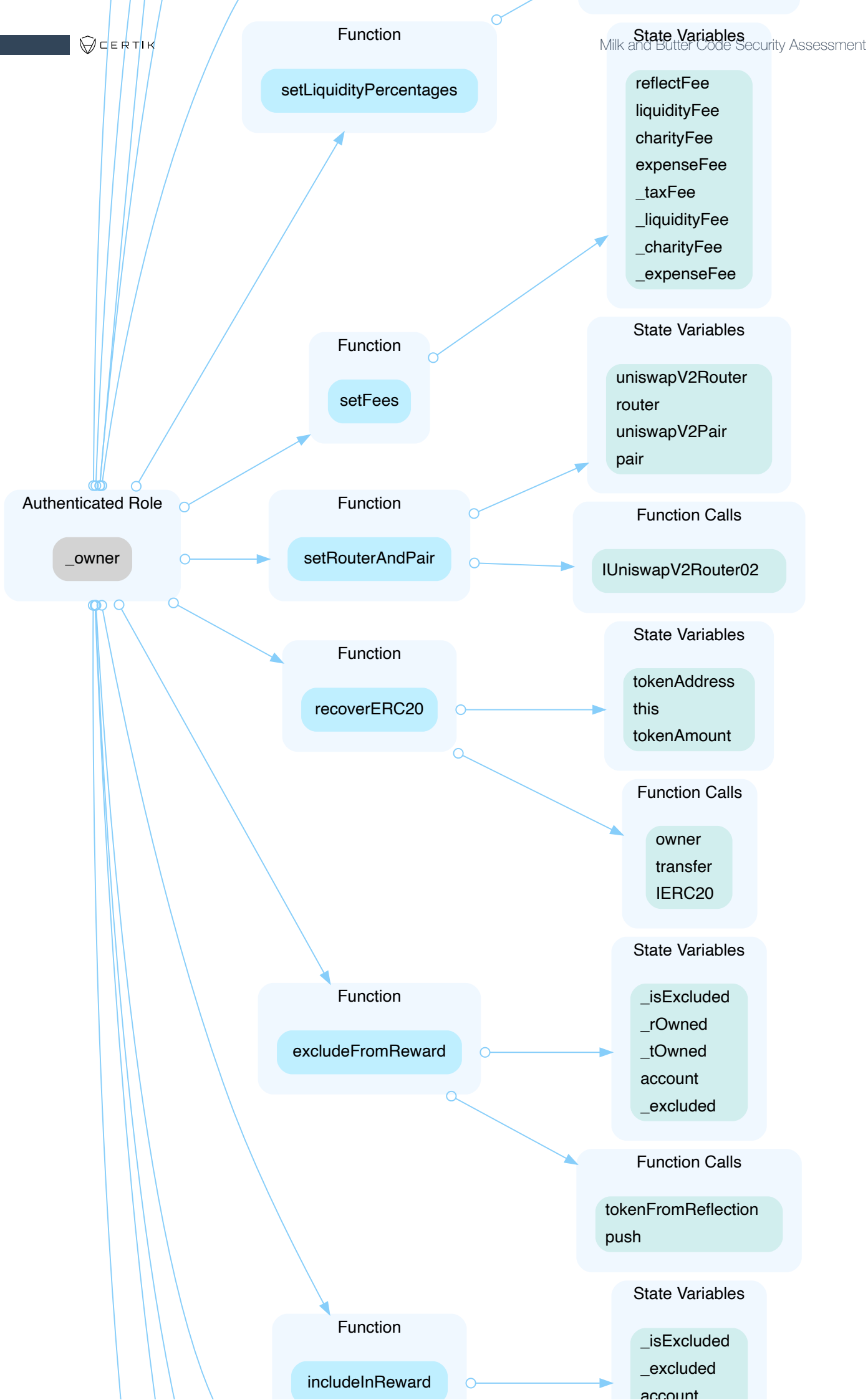
Category	Severity	Location	Status
Centralization / Privilege	Major	projects/MilkToken/contracts/MilkAndButter.sol: 769~772, 774~777, 779~782, 784~787, 789~792, 794~800, 802~809, 811~815, 817~820, 942~950, 952~963, 977~979, 981~983, 985~988, 1136~1139	Acknowledged

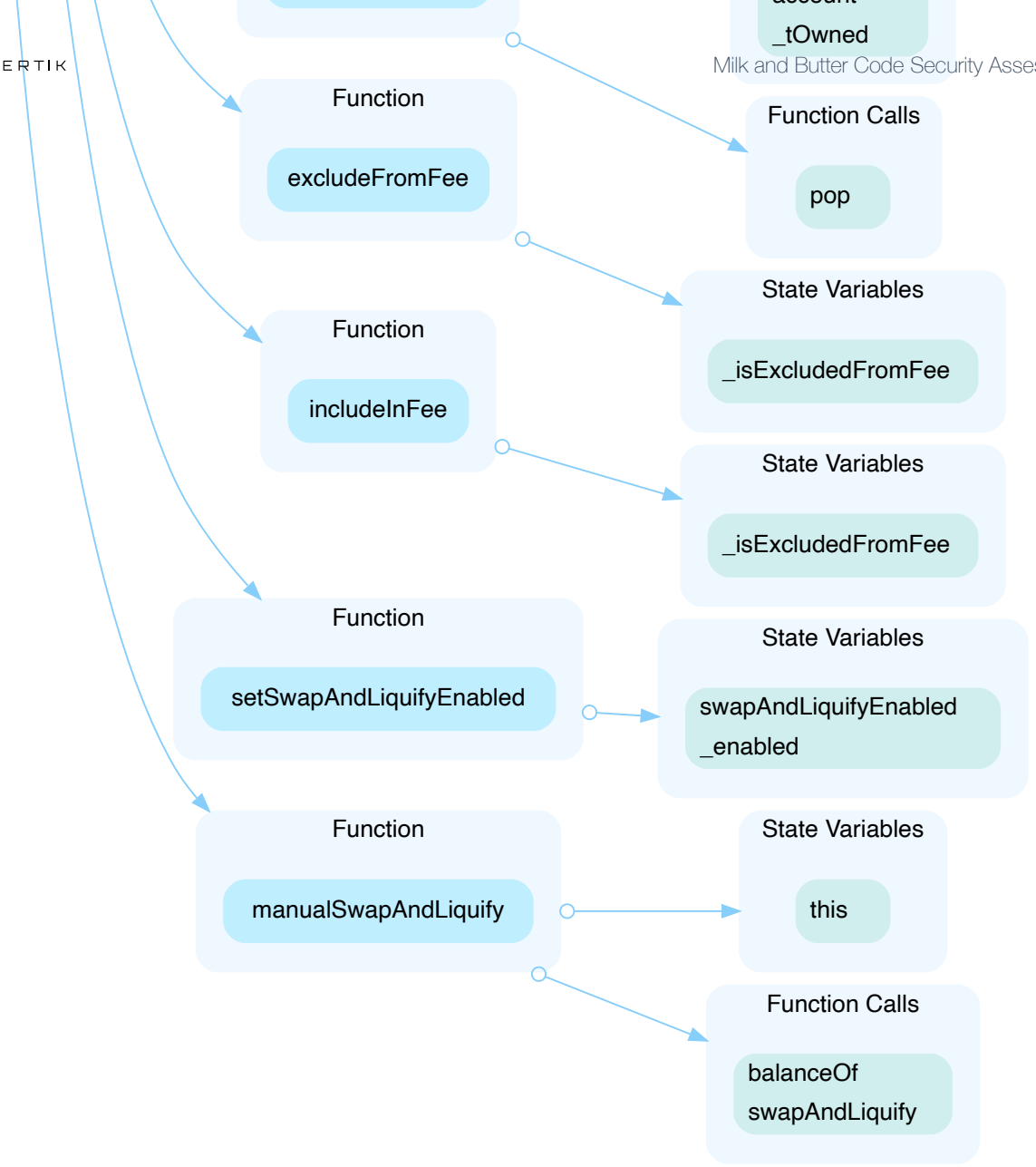
Description

In the contract, `MilkAndButter`, 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.




Function
setLiquidityPercentages
State Variables
reflectFee
liquidityFee
charityFee
expenseFee
_taxFee
_liquidityFee
_charityFee
_expenseFee
Function
setFees
State Variables
uniswapV2Router
router
uniswapV2Pair
pair
Authenticated Role
_owner
Function
setRouterAndPair
Function Calls
IUniswapV2Router02
Function
recoverERC20
State Variables
tokenAddress
this
tokenAmount
Function Calls
owner
transfer
IERC20
Function
excludeFromReward
State Variables
_isExcluded
_rOwned
_tOwned
account
_excluded
Function Calls
tokenFromReflection
push
Function
includeInReward
State Variables
_isExcluded
_excluded
account



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

[Milk and Butter Team]: The Milk and Butter team acknowledges the centralization risk associated with the _owner account. As such, the team has made certain that the account keys are kept in a secure manner and the team will move forward with the recommendation of moving these privileged functions to a Mutltisign contract.

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

Category	Severity	Location	Status
Gas Optimization	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 676, 680, 681, 682, 690, 691, 700	Acknowledged

Description

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

Recommendation

We recommend to declare these variables as `constant`.

MAB-03 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 769~772, 774~777, 779~782, 784~787, 789~792, 794~800, 802~809, 811~815, 817~820 , 942~950, 952~963, 977~979, 981~983, 1136~1139	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.

MAB-04 | Improper Usage Of `public` And `external` Type

Category	Severity	Location	Status
Gas Optimization	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 432~436, 802~809, 942~950, 747~750, 892~896, 757~760, 752~755, 789~792, 903~906, 869~871, 762~765, 774~777, 981~983, 857~859, 977~979, 1136~1139, 727~730, 779~782, 985~988, 794~800, 887~890, 916~923, 925~934, 898~901, 912~914, 861~863, 769~772, 811~815, 817~820, 908~910, 883~885, 1078~1080, 784~787, 865~867	Acknowledged

Description

`public` functions that are never called by the contract could be declared as `external`. `external` functions are more efficient than `public` functions.

Recommendation

Consider using the external attribute for public functions that are never called within the contract.

MAB-05 | Typos In The Contract

Category	Severity	Location	Status
Coding Style	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 829, 990	Acknowledged

Description

There are several typos in the code and comments.

1. In the following code snippet, `tokensIntoLiqudity` should be `tokensIntoLiquidity`.

```
event SwapAndLiquify(
        uint256 tokensSwapped,
        uint256 ethReceived,
        uint256 tokensIntoLiqudity
    );
```

2. `recieve` should be `receive` and `swaping` should be `swapping` in the line of comment `//to recieve ETH from uniswapV2Router when swaping`.

Recommendation

We advise the team to consider correcting all typos in the contract.

MAB-06 | Incorrect Error Message

Category	Severity	Location	Status
Logical Issue	Minor	projects/MilkToken/contracts/MilkAndButter.sol: 953	Acknowledged

Description

The error message in `require(_isExcluded[account], "Account is already excluded")` does not describe the error correctly.

Recommendation

The message "Account is already excluded" can be changed to "Account is not excluded" .

Alleviation

[Milk and Butter tTeam]: The team acknowledges that this error message is indeed incorrect, however, since the code-base has already been deployed the issue cannot be fixed

MAB-07 | Contract Gains Non-withdrawable BNB Via The `swapAndLiquify` Function

Category	Severity	Location	Status
Logical Issue	Medium	projects/MilkToken/contracts/MilkAndButter.sol: 1141	Acknowledged

Description

The `swapAndLiquify` function converts half of the `tokensToLiquify` MB tokens to BNB. The other half of MB tokens and part of the converted BNB are deposited into the MB-BNB pool on pancakeswap as liquidity. For every `swapAndLiquify` function call, a small amount of BNB leftover in the contract. This is because the price of MB drops after swapping the first half of MB tokens into BNBs, and the other half of MB tokens require less than the converted BNB to be paired with it when adding liquidity. The contract doesn't appear to provide a way to withdraw those BNB, and they will be locked in the contract forever.

Recommendation

It's not ideal that more and more BNB are locked into the contract over time. The simplest solution is to add a `withdraw` function in the contract to withdraw BNB. Other approaches that benefit the MB token holders can be:

- Distribute BNB to MB token holders proportional to the amount of token they hold.
- Use leftover BNB to buy back MB tokens from the market to increase the price of MB.

Alleviation

[Milk and Butter Team]: The team acknowledges that the contract gains non-withdrawable BNB. However, this amount seems to be very small upon observing the current behavior of the contract and the code-base cannot be edited

MAB-08 | Return Value Not Handled

Category	Severity	Location	Status
Volatile Code	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 1227~1234, 1236~1243, 1245~1252	Acknowledged

Description

The return values of function `addLiquidityETH` are not properly handled.

```
        uniswapV2Router.addLiquidityETH{value: ethToLock}(
            address(this),
            tokensToLock,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            lockedLiquidity(),
            block.timestamp
        );

        uniswapV2Router.addLiquidityETH{value: ethToManager}(
            address(this),
            tokensToManager,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            liquidityManager(),
            block.timestamp
        );

        uniswapV2Router.addLiquidityETH{value: ethToBurn}(
            address(this),
            tokensToBurn,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            burn(),
            block.timestamp
        );
```

Recommendation

We advise the team to consider using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic.

MAB-09 | The Purpose Of Function `deliver`

Category	Severity	Location	Status
Control Flow	Informational	projects/MilkToken/contracts/MilkAndButter.sol: 916~923	Acknowledged

Description

The function `deliver` can be called by anyone. It accepts an uint256 number parameter `tAmount`. The function reduces the MB token balance of the caller by `rAmount`, which is `tAmount` reduces the transaction fee. Then, the function adds `tAmount` to variable `_tFeeTotal`, which represents the contract's total transaction fee.

Recommendation

We wish the team could explain more on the purpose of having such functionality.

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.