CERTIK

Preliminary Comments

petcoin.love

Oct 29th, 2021

Table of Contents

Summary

This report has been prepared for petcoin.love to discover issues and vulnerabilities in the source code of the petcoin.love 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	petcoin.love
Platform	BSC
Language	Solidity
Codebase	https://testnet.bscscan.com/address/0xdf4fac13d29de69facfec55751da55ad59185b21#code
Commit	

Audit Summary

Delivery Date	Oct 29, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

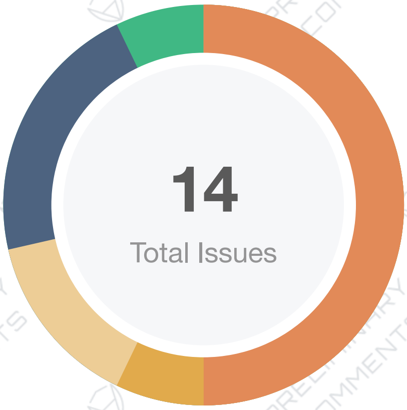
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● Major	7	7	0	0	0	0
● Medium	1	1	0	0	0	0
● Minor	2	2	0	0	0	0
● Informational	3	3	0	0	0	0
● Discussion	1	1	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
PCP	PetCoinToken.sol	470889624b1a4574001a129979a6453a10cb5980a576c94e57355267bb967240

Findings



■	**Critical**	**0** (0.00%)
■	**Major**	**7** (50.00%)
■	**Medium**	**1** (7.14%)
■	**Minor**	**2** (14.29%)
■	**Informational**	**3** (21.43%)
■	**Discussion**	**1** (7.14%)

ID	Title	Category	Severity	Status
PCP-01	Centralization Risk	**Centralization / Privilege**	● **Major**	(!) Pending
PCP-02	Centralization Risk	**Centralization / Privilege**	● **Major**	(!) Pending
PCP-03	Initial Token Distribution	**Centralization / Privilege**	● **Major**	(!) Pending
PCP-04	Return value not handled	Volatile Code	● Informational	(!) Pending
PCP-05	Centralized risk in `addLiquidity`	**Centralization / Privilege**	● **Major**	(!) Pending
PCP-06	Contract gains non-withdrawable BNB via the `swapAndLiquify` function	Logical Issue	● Major	(!) Pending
PCP-07	Third Party Dependencies	Volatile Code	● Minor	(!) Pending
PCP-08	Limited Effect to Prevent the Selling and Buying	Logical Issue	● Medium	(!) Pending
PCP-09	Valid value of `startBlockSwap`	Logical Issue	● Discussion	(!) Pending
PCP-10	Fee Collectors	**Centralization / Privilege**	● **Major**	(!) Pending
PCP-11	Unused Variable	Gas Optimization	● Informational	(!) Pending
PCP-12	Variables that could be declared as `constant`	Gas Optimization	● Informational	(!) Pending
PCP-13	Wrong Amount To Transfer	Logical Issue	● Major	(!) Pending

ID	Title	Category	Severity	Status
PCP-14	Usage of `transfer()` for sending Ether	Volatile Code	● Minor	ⓘ Pending

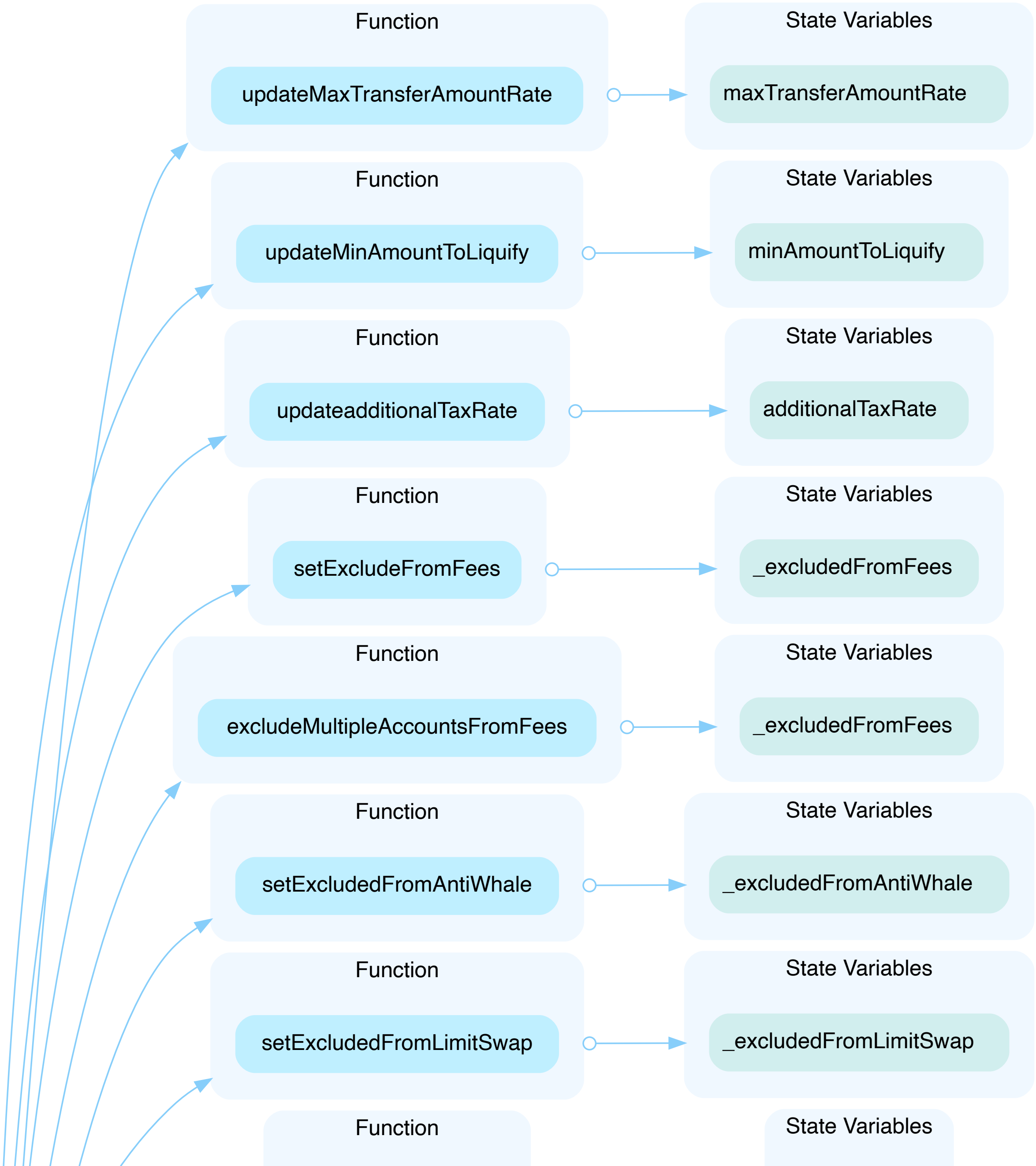
PCP-01 | Centralization Risk

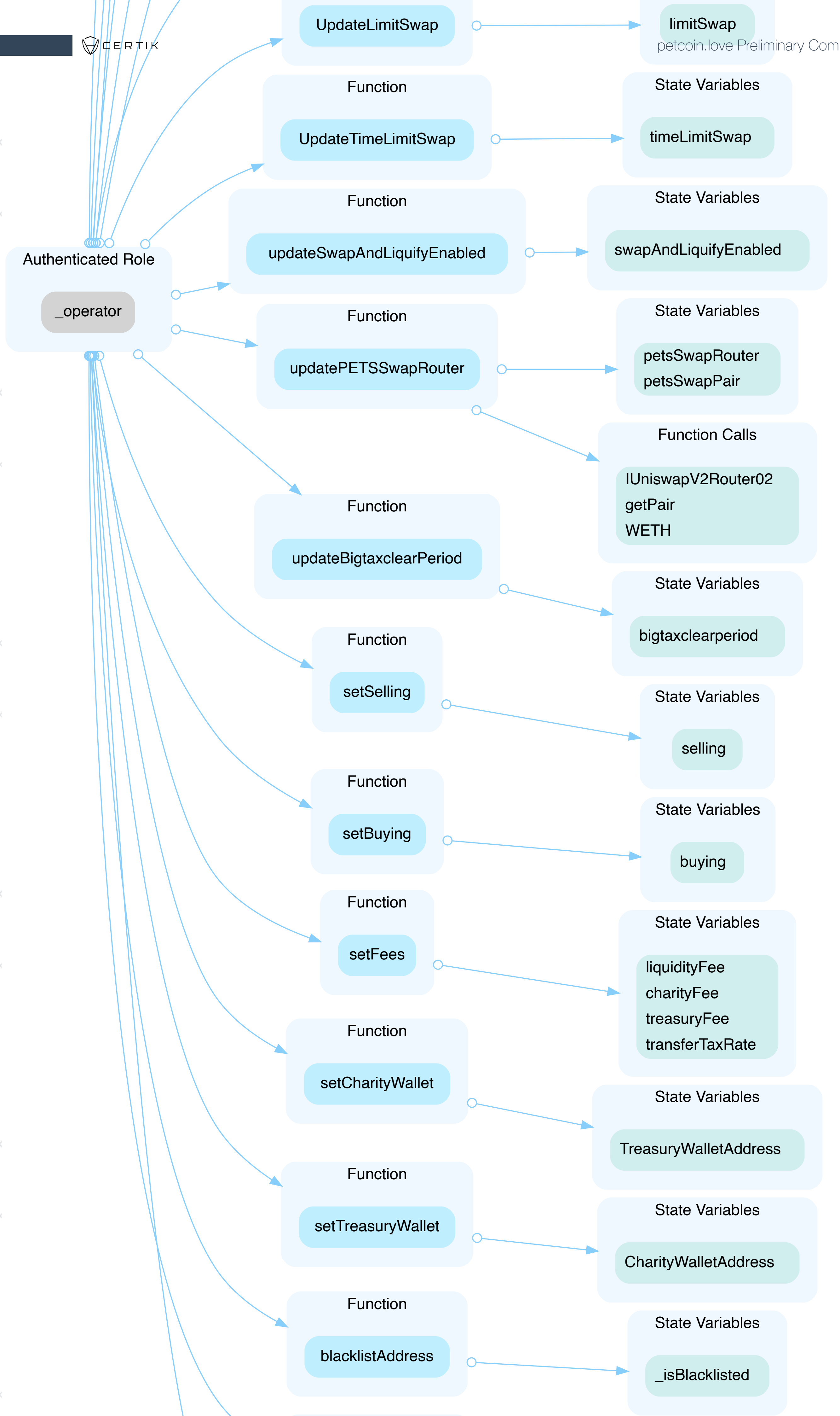
Category	Severity	Location	Status
Centralization / Privilege	● Major	PetCoinToken.sol: 1531~1535, 1540~1543, 1546~1549, 1554~1559, 1564~1569, 1574~1577, 1582~1585, 1590~1593, 1598~1602, 1607~1610, 1615~1620, 1622~1626, 1645~1647, 1652~1654, 1659~1666, 1671~1673, 1678~1680, 1685~1687, 1692~1696, 1700~1707	ⓘ Pending

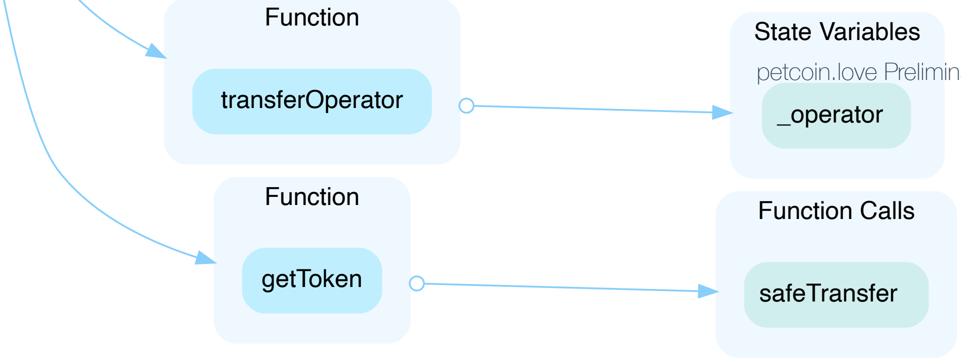
Description

In the contract, `PetCoinToken`, the role, `_operator`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_operator` may allow the hacker to take advantage of this.




UpdateLimitSwap
limitSwap
Function
UpdateTimeLimitSwap
State Variables
timeLimitSwap
Function
updateSwapAndLiquifyEnabled
State Variables
swapAndLiquifyEnabled
Authenticated Role
_operator
Function
updatePETSSwapRouter
State Variables
petsSwapRouter
petsSwapPair
Function Calls
IUniswapV2Router02
getPair
WETH
Function
updateBigtaxclearPeriod
State Variables
bigtaxclearperiod
Function
setSelling
State Variables
selling
Function
setBuying
State Variables
buying
Function
setFees
State Variables
liquidityFee
charityFee
treasuryFee
transferTaxRate
Function
setCharityWallet
State Variables
TreasuryWalletAddress
Function
setTreasuryWallet
State Variables
CharityWalletAddress
Function
blacklistAddress
State Variables
_isBlacklisted



Recommendation

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 to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

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

PCP-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	PetCoinToken.sol: 1630~1640	ⓘ Pending

Description

In the contract `PetCoinToken`, the role `owner` has the authority over the following function:

- `UpdateStartBlockSwap(uint256 _block)`

Any compromise to the `owner` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

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

PCP-03 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	● Major	PetCoinToken.sol: 1322	ⓘ Pending

Description

`10000000000 * (10**9)` of the `PETS` tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute `PETS` tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process.

PCP-04 | Return value not handled

Category	Severity	Location	Status
Volatile Code	● Informational	PetCoinToken.sol: 1476~1483	Pending

Description

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

```
        petsSwapRouter.addLiquidityETH{value: ethAmount}(
            address(this),
            tokenAmount,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            owner(),
            block.timestamp
        );
```

Recommendation

We recommend 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.

PCP-05 | Centralized risk in `addLiquidity`

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	PetCoinToken.sol: 1481	ⓘ Pending

Description

```
// add the liquidity
petsSwapRouter.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        owner(),
        block.timestamp
     );
```

The `addLiquidity` function calls the `petsSwapRouter.addLiquidityETH` function with the `to` address specified as `owner()` for acquiring the generated LP tokens from the `PETS-BNB` pool. As a result, over time the `_owner` address will accumulate a significant portion of LP tokens. 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 advise the `to` address of the `petsSwapRouter.addLiquidityETH` function call to be replaced by the contract itself, i.e. `address(this)`, and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the `_owner` account is compromised. In general, 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 a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

PCP-06 | Contract gains non-withdrawable BNB via the `swapAndLiquify` function

Category	Severity	Location	Status
Logical Issue	● Major	PetCoinToken.sol: 1426	ⓘ Pending

Description

The `swapAndLiquify` function converts half of the `contractTokenBalance` PETS tokens to BNB. The other half of PETS tokens and part of the converted BNB are deposited into the PETS-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 PETS drops after swapping the first half of PETS tokens into BNBs, and the other half of PETS 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 PETS token holders can be:

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

PCP-07 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	● Minor	PetCoinToken.sol: 1155	ⓘ Pending

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 `PetCoinToken` 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.

PCP-08 | Limited Effect to Prevent the Selling and Buying

Category	Severity	Location	Status
Logical Issue	● Medium	PetCoinToken.sol: 1250	ⓘ Pending

Description

It is noted that the restrictions to prevent the selling and the buying are limited to the specified `petsSwapPair` pair. Is that designed as expected?

PCP-09 | Valid value of `startBlockSwap`

Category	Severity	Location	Status
Logical Issue	● Discussion	PetCoinToken.sol: 1192	ⓘ Pending

Description

Both current block number of Ethereum and BSC are far less the value of `startBlockSwap`. Is that designed as expected?

PCP-10 | Fee Collectors

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	PetCoinToken.sol: 1383, 1404, 1407	ⓘ Pending

Description

There are two fee collectors, i.e. `TreasuryWalletAddress` and `CharityWalletAddress`, over time, the two accounts would gain much fee.

Recommendation

In general, 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.

PCP-11 | Unused Variable

Category	Severity	Location	Status
Gas Optimization	● Informational	PetCoinToken.sol: 1160~1162	ⓘ Pending

Description

The state variables `LiquidityPoolWalletAddress`, `PublicSaleWalletAddress`, and `ReserveWalletAddress` are unused.

Recommendation

Consider removing those unused variables.

PCP-12 | Variables that could be declared as `constant`

Category	Severity	Location	Status
Gas Optimization	● Informational	PetCoinToken.sol: 1180	ⓘ Pending

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`.

PCP-13 | Wrong Amount To Transfer

Category	Severity	Location	Status
Logical Issue	● Major	PetCoinToken.sol: 1700~1707	ⓘ Pending

Description

In the function `getToken`, the amount to transfer should be the input param `_amount` while it actually is the local variable `amount`:

```
        uint256 amount = _token.balanceOf(address(this));
        if( _amount > amount){amount = _amount;}
        _token.safeTransfer(_recipient, amount);
```

Recommendation

Consider refactoring the function `getToken`, for example:

```
        uint256 amount = _token.balanceOf(address(this));
        if( _amount > amount){_amount = amount;}
        _token.safeTransfer(_recipient, _amount);
```

PCP-14 | Usage of `transfer()` for sending Ether

Category	Severity	Location	Status
Volatile Code	● Minor	PetCoinToken.sol: 1421	ⓘ Pending

Description

After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` or `.send()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case the linked statements are meant to be able to transfer funds to other contracts instead of EOAs.

Recommendation

We advise that the linked `.transfer()` and `.send()` calls are substituted with the utilization of the `sendValue()` function from the `Address.sol` implementation of OpenZeppelin either by directly importing the library or copying the linked code.

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.

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.