CERTIK

Security Assessment

Metaversefacemaker

Apr 20th, 2022

Table of Contents

Summary

This report has been prepared for Metaversefacemaker to discover issues and vulnerabilities in the source code of the Metaversefacemaker 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	Metaversefacemaker
Platform	BSC
Language	Solidity
Codebase	https://github.com/MetaFaceNft/mefa
Commit	fb4eb2b83d0bc083c25701f681399ad32d6a2a07

Audit Summary

Delivery Date	Apr 20, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

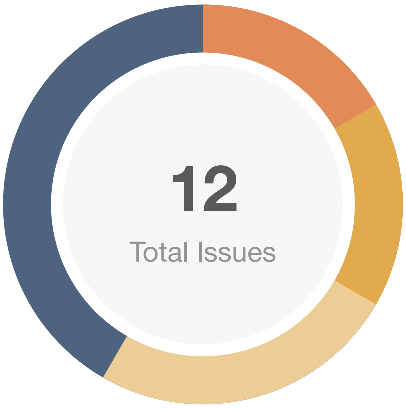
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	2	0	0	2	0	0	0
● Medium	2	0	0	1	0	0	1
● Minor	3	0	0	2	0	0	1
● Informational	5	0	0	4	0	0	1
● Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
CCK	Contract.sol	94d9c82f7ed9f8b8606cadb177d9c858c3369dd05ec477e9ff0aab7aecef4c15
CKP		

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**2** (16.67%)
🟨 **Medium**	**2** (16.67%)
🟨 **Minor**	**3** (25.00%)
🟦 **Informational**	**5** (41.67%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
MEFA-01	3rd Party Dependencies	Control Flow	● Minor	ⓘ Acknowledged
CCK-01	Centralization Risk In Contract.sol	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
CCK-02	Centralized Risk In `addLiquidity`	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
CCK-03	Initial Token Distribution	**Centralization / Privilege**	● **Medium**	✓ Resolved
CCK-04	Contract Gains Non-withdrawable BNB Via The `swapAndLiquify` Function	Logical Issue	● Medium	ⓘ Acknowledged
CCK-05	Incorrect Error Message	Logical Issue	● Minor	ⓘ Acknowledged
CCK-06	Unknown Implementation Of `onPreTransferCheck`	Volatile Code	● Minor	✓ Resolved
CCK-07	Missing Emit Events	Coding Style	● Informational	ⓘ Acknowledged
CCK-08	Variables That Could Be Declared As Immutable	Gas Optimization	● Informational	ⓘ Acknowledged
CCK-09	Tautology Or Contradiction	Gas Optimization	● Informational	ⓘ Acknowledged
CCK-10	Redundant Code	Logical Issue	● Informational	ⓘ Acknowledged
CCK-11	The Purpose Of Function `deliver`	Control Flow	● Informational	✓ Resolved

MEFA-01 | 3rd Party Dependencies

Category	Severity	Location	Status
Control Flow	● Minor	Global	ⓘ Acknowledged

Description

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

Recommendation

We understand that the business logic of the Metaversefacemaker protocol requires the interaction PancakeSwap protocol for adding liquidity to the MEFA-BNB pool and swap tokens. We encourage the team to constantly monitor the statuses of those 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The Metaverse Face team acknowledged this finding.

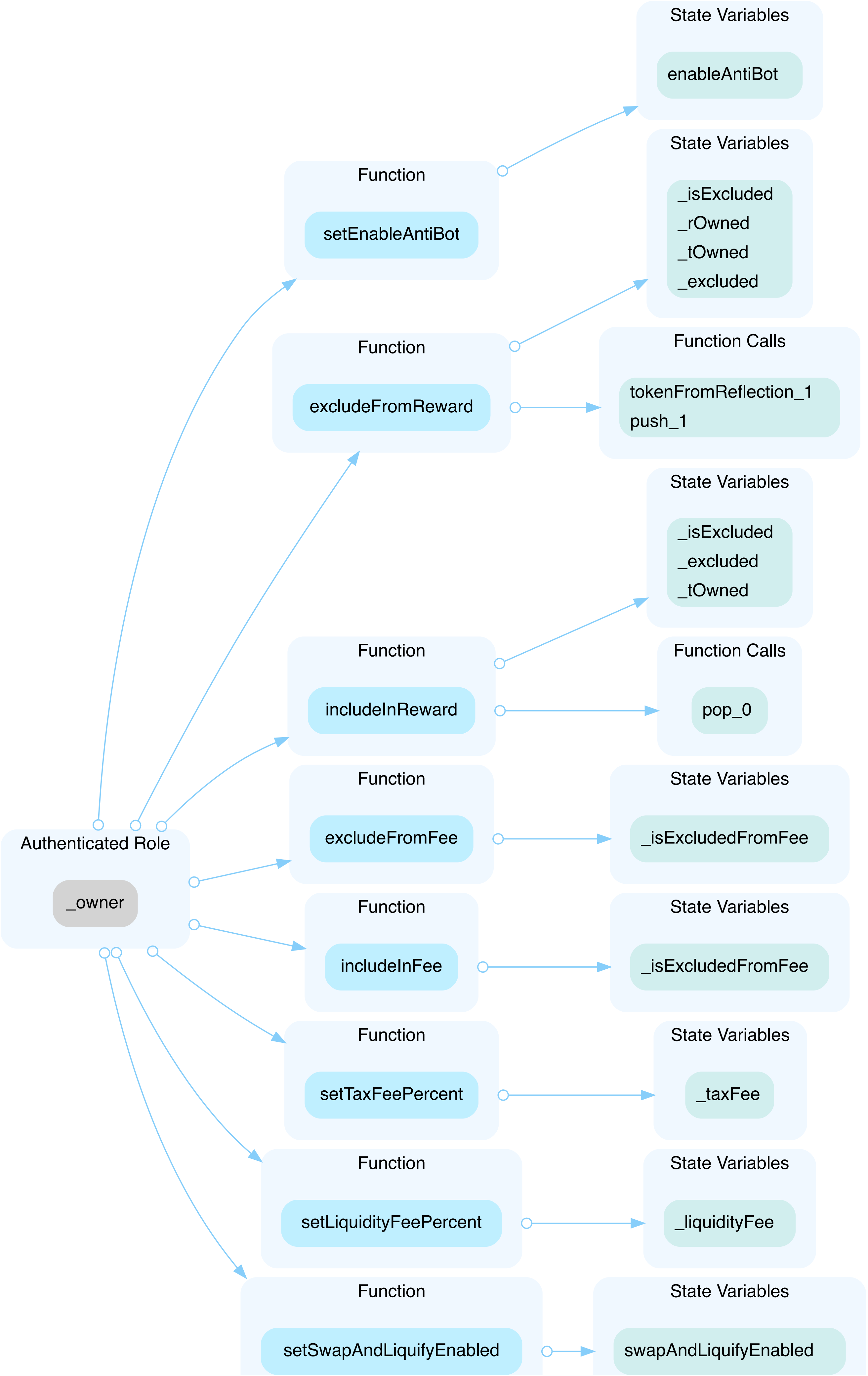
CCK-01 | Centralization Risk In Contract.sol

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Contract.sol: 169~171, 177~180, 1079~1081, 1225~1233, 1235~1246 , 1272~1274, 1276~1278, 1280~1283, 1285~1294, 1296~1299	ⓘ Acknowledged

Description

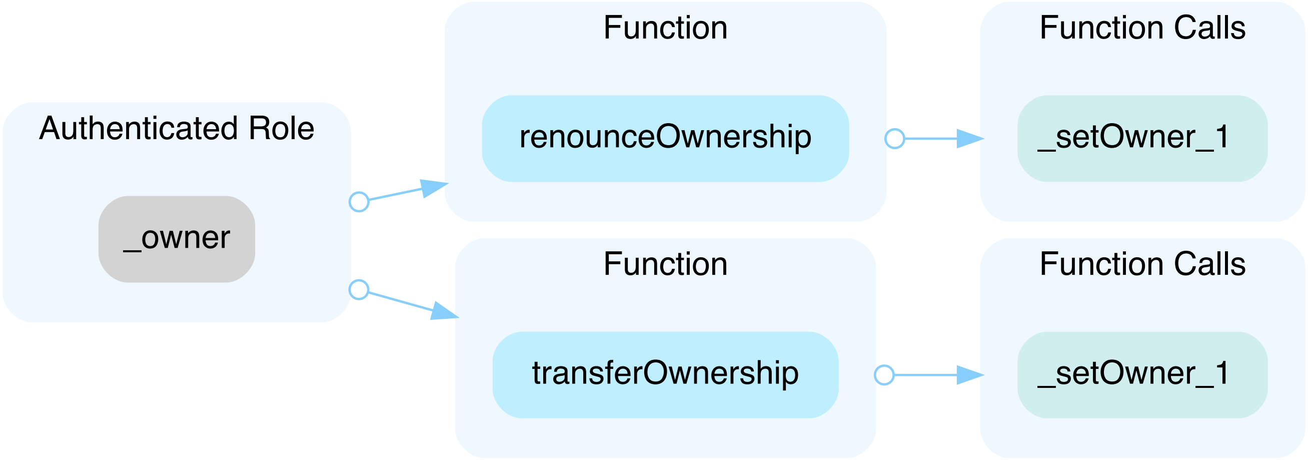
In the contract `AntiBotLiquidityGeneratorToken` the role `_owner` has authority over the functions shown in the diagram below.

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


State Variables
enableAntiBot
State Variables
_isExcluded
_rOwned
_tOwned
_excluded
Function
setEnableAntiBot
Function
excludeFromReward
Function Calls
tokenFromReflection_1
push_1
State Variables
_isExcluded
_excluded
_tOwned
Function
includeInReward
Function Calls
pop_0
Function
excludeFromFee
State Variables
_isExcludedFromFee
Authenticated Role
_owner
Function
includeInFee
State Variables
_isExcludedFromFee
Function
setTaxFeePercent
State Variables
_taxFee
Function
setLiquidityFeePercent
State Variables
_liquidityFee
Function
setSwapAndLiquifyEnabled
State Variables
swapAndLiquifyEnabled

In the contract `Ownable` 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.



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

The Metaverse Face team acknowledges this finding and they will renounce ownership upon deployment.

CCK-02 | Centralized Risk In `addLiquidity`

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Contract.sol: 1581	ⓘ Acknowledged

Description

```
1575 // add the liquidity
1576 uniswapV2Router.addLiquidityETH{value: ethAmount}(
1577     address(this),
1578     tokenAmount,
1579     0, // slippage is unavoidable
1580     0, // slippage is unavoidable
1581     owner(),
1582     block.timestamp
1583 );
```

The `addLiquidity` function calls the `uniswapV2Router.addLiquidityETH` function with the `to` address specified as `owner()` for acquiring the generated LP tokens from the `MEFA-WBNB` 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 for the project as a whole.

Recommendation

We advise the `to` address of the `uniswapV2Router.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 single point of failure due to the private key;
- Introduction of a DAO / governance / voting module to increase transparency and user involvement.

Alleviation

The Metaverse Face team acknowledges this finding and they will renounce ownership upon deployment.

CCK-03 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	● **Medium**	Contract.sol: 1053	✓ Resolved

Description

All of the `MEFA` tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute all 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

[Metaverse Face Team]: We use bubble maps as tokenomics. People can follow this section transparently. All major wallets are owned by exchanges. Also, main contract and charity addresses and special words are definitely well implemented.

https://bubbles.moonlighttoken.com/bsc/token/0x6ad0f087501eee603aeda0407c52864bc7f83322

CCK-04 | Contract Gains Non-withdrawable BNB Via The `swapAndLiquify` Function

Category	Severity	Location	Status
Logical Issue	● Medium	Contract.sol: 1530	ⓘ Acknowledged

Description

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

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

Alleviation

The Metaverse Face team acknowledged this finding.

CCK-05 | Incorrect Error Message

Category	Severity	Location	Status
Logical Issue	● Minor	Contract.sol: 1236	ⓘ 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

The Metaverse Face team acknowledged this finding, and given the deployed contract cannot be updated, decided to retain the code base unchanged.

CCK-06 | Unknown Implementation Of `onPreTransferCheck`

Category	Severity	Location	Status
Volatile Code	● Minor	Contract.sol: 1496	✓ Resolved

Description

The method `onPreTransferCheck` in `pinkAntiBot` is unknown implementation.

```
1495  if (enableAntiBot) {
1496      pinkAntiBot.onPreTransferCheck(from, to, amount);
1497  }
```

The scope of the audit treats 3rd party entities as black boxes and assumes 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 this protocol requires interaction with these functions. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

[Metaverse Face Team]: We do pre-sale on Pinksale. This system was so that bots could not participate in the pre-sale. Other than that it has never been used.

CCK-07 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	Contract.sol: 169~171, 177~180, 1079~1081, 1225~1233, 1235~1246, 1272~1274, 1276~1278, 1280~1283, 1285~1294	ⓘ 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

The Metaverse Face team acknowledged this finding, and given the deployed contract cannot be updated, decided to retain the code base unchanged.

CCK-08 | Variables That Could Be Declared As Immutable

Category	Severity	Location	Status
Gas Optimization	● Informational	Contract.sol: 952, 958, 970, 971, 976	ⓘ Acknowledged

Description

The linked variables assigned in the constructor can be declared as `immutable`. Immutable state variables can be assigned during contract creation but will remain constant throughout the lifetime of a deployed contract. A big advantage of immutable variables is that reading them is significantly cheaper than reading from regular state variables since they will not be stored in storage.

Recommendation

We recommend declaring these variables as immutable. Please note that the `immutable` keyword only works in Solidity version `v0.6.5` and up.

Alleviation

The Metaverse Face team acknowledged this finding, and given the deployed contract cannot be updated, decided to retain the code base unchanged.

CCK-09 | Tautology Or Contradiction

Category	Severity	Location	Status
Gas Optimization	● Informational	Contract.sol: 1008, 1009~1012, 1013~1016, 1281, 1289~1292	ⓘ Acknowledged

Description

Comparisons that are always true or always false may be incorrect or unnecessary.

```
1008  require(taxFeeBps_ >= 0 && taxFeeBps_ <= 10**4, "Invalid tax fee");
```

Since the variable `taxFeeBps_` is of type uint16, it must satisfy greater than or equal to 0.

Recommendation

We recommend fixing the incorrect comparison by changing the value type or the comparison operator.

Alleviation

The Metaverse Face team acknowledged this finding, and will renounce the ownership upon deployment.

CCK-10 | Redundant Code

Category	Severity	Location	Status
Logical Issue	● Informational	Contract.sol: 1600	ⓘ Acknowledged

Description

The condition `!_isExcluded[sender] && !_isExcluded[recipient]` can be included in `else` .

Recommendation

The following code can be removed:

```
... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
    _transferStandard(sender, recipient, amount);
} ...
```

Alleviation

The Metaverse Face team acknowledged this finding, and given the deployed contract cannot be updated, decided to retain the code base unchanged.

CCK-11 | The Purpose Of Function `deliver`

Category	Severity	Location	Status
Control Flow	● Informational	Contract.sol: 1185~1195	✓ Resolved

Description

The function `deliver()` can be called by anyone. It accepts an uint256 number parameter `tAmount`. The function reduces the MEFA 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. We wish the team could explain more on the purpose of having such functionality.

Recommendation

Please review this function to ensure it meets the design intent.

Alleviation

[Metaverse Face Team]: Since the contract is a pinksale contract, we think this is the case.

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.

CERTIK