CERTIK

Security Assessment

Meerdefi

Apr 26th, 2022

Table of Contents

Summary

This report has been prepared for Meerdefi to discover issues and vulnerabilities in the source code of the Meerdefi 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	Meerdefi
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/zhangyi999/meerco
Commit	2c1c79c2948ef2aacddaa34b003fd777ddcb6282

Audit Summary

Delivery Date	Apr 26, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

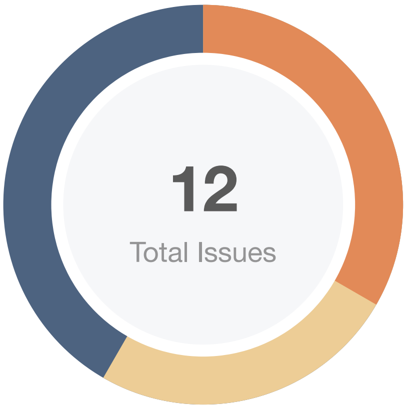
Vulnerability Summary

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

Audit Scope

ID	File	SHA256 Checksum
RCK	Relation.sol	663a548c2059dd21b8c2041ce47e984082d12f345a44ef31f21377c299d1dc80
SLP	stake/StakeLP.sol	66f90f9e44dab7535cfc663ae237ce5f3a057aaf1309446df913cce9a8b7b945
BMC	BMeer.sol	9b6c6a3abfa912c944b4f341ffdaa3c82b893b20f5978196954281147eb5bcc1

Findings



🟥	**Critical**	**0** (0.00%)
🟧	**Major**	**4** (33.33%)
🟨	**Medium**	**0** (0.00%)
🟨	**Minor**	**3** (25.00%)
🟦	**Informational**	**5** (41.67%)
🟩	**Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
MEER-01	Third Party Dependencies	Logical Issue	● Minor	ⓘ Acknowledged
BMC-01	Centralization Risk In BMeer.sol	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
CKP-01	Unlocked Compiler Version	Language Specific	● Informational	ⓘ Acknowledged
CKP-02	Missing Emit Events	Language Specific	● Informational	ⓘ Acknowledged
RCK-01	Centralization Risk In Relation.sol	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
SLP-01	Centralization Risk In StakeLP.sol	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
SLP-02	Lack Of Asset Transfer At UnStake() Function	Logical Issue	● Major	✓ Resolved
SLP-03	Missing Check For Unlock Time	Logical Issue	● Minor	ⓘ Acknowledged
SLP-04	Lack Of Access Control On `claimFor()`	Logical Issue	● Minor	ⓘ Acknowledged
SLP-05	Unused Return Value	Volatile Code	● Informational	ⓘ Acknowledged
SLP-06	Typo And Incorrect Comments	Coding Style	● Informational	ⓘ Acknowledged
SLP-07	Number Of Denominator EXP_RATE	Logical Issue	● Informational	ⓘ Acknowledged

MEER-01 | Third Party Dependencies

Category	Severity	Location	Status
Logical Issue	● Minor		ⓘ Acknowledged

Description

The contract is serving as the underlying entity to interact with the third parties, the following is a list of contracts and interfaces. 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.

- `ITrimLp`
- `Iminer`
- `Iburn`
- `usd`
- `meer`
- `swapLP`
- `safeToken`

Recommendation

We understand that business logic needs to be interactive. We encourage the team to constantly monitor the status of third parties to mitigate the side effects when unexpected activity is observed.

Alleviation

The MeerDefi team acknowledged this finding.

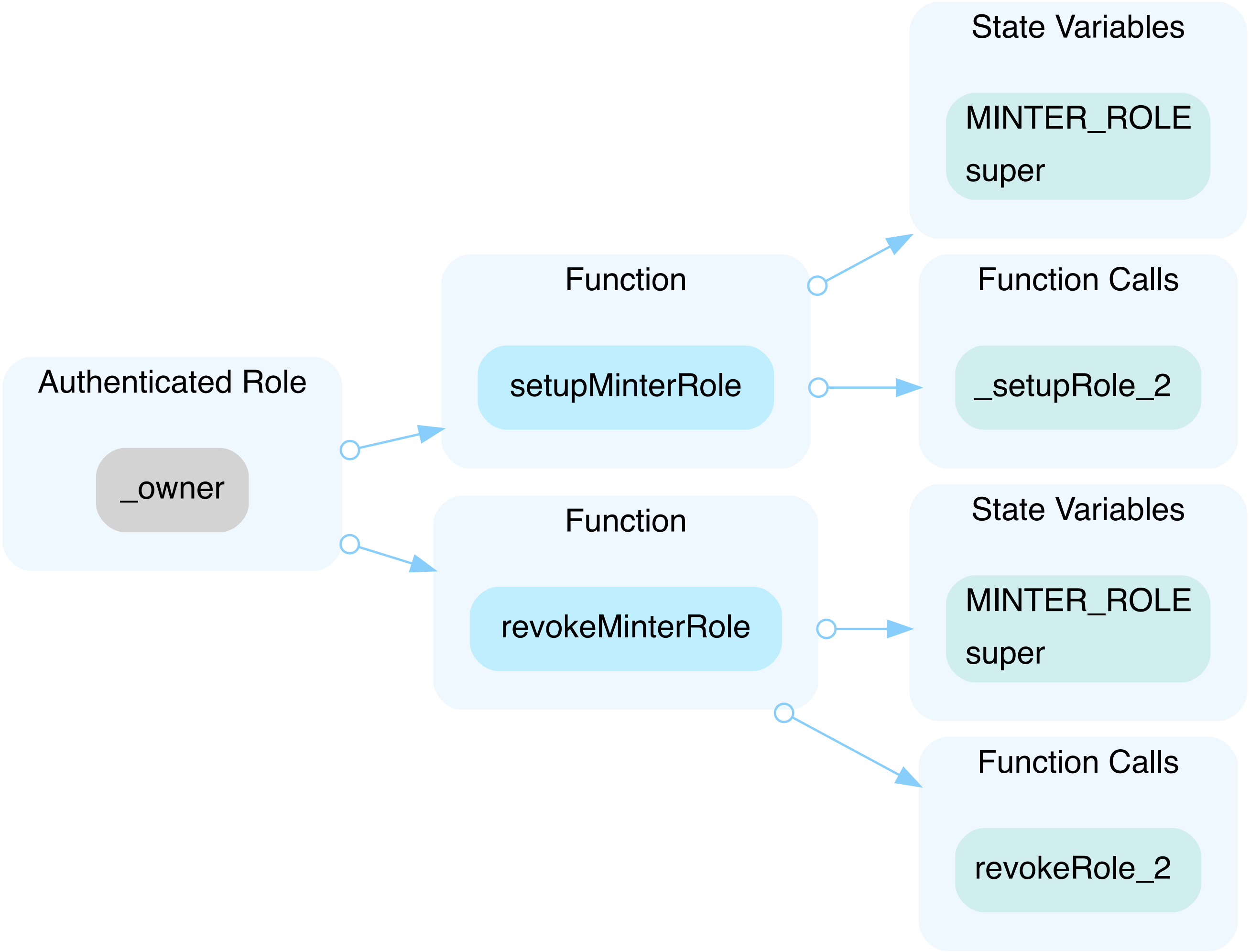
BMC-01 | Centralization Risk In BMeer.sol

Category	Severity	Location	Status
Centralization / Privilege	● Major	BMeer.sol: 23~25, 27~29	ⓘ Acknowledged

Description

In the contract `MinterAccess` 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.



In the contract `BMeer`, the role `Minter` has authority over the following functions:

- mint(): Add `amount` number of `MEER` tokens to any given address

Any compromise to the `Minter` account may allow a 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 MeerDefi team acknowledged this finding.

CKP-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	● Informational	stake/StakeLP.sol: 2; BMeer.sol: 2; Relation.sol: 2	ⓘ Acknowledged

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to different compiler versions. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.7` the contract should contain the following line:

```
pragma solidity 0.8.7;
```

Alleviation

The MeerDefi team acknowledged this finding.

CKP-02 | Missing Emit Events

Category	Severity	Location	Status
Language Specific	● Informational	stake/StakeLP.sol: 22~24, 177, 187; Relation.sol: 28	ⓘ Acknowledged

Description

One or more state changes do not emit events to pass the changes out of the chain.

File: projects/Relation.sol (Line 28, Function `Relation.setMaxNmber`)

```
        maxNmbrella =  _maxNmbrella;
```

File: projects/stake/StakeLP.sol (Line 177, Function `StakeMeerLp.setShareNetRate`)

```
        shareNetRateEPX_RATE = _rate;
```

File: projects/stake/StakeLP.sol (Line 187, Function `StakeMeerLp.setMintRateEPX_RATE`)

```
        mintMeerRate_EPX_RATE = _mintRate;
```

File: projects/stake/StakeLP.sol (Line 23, Function `Owner.setOwner`)

```
        _owner = _owner_;
```

Recommendation

We recommend declaring and emitting corresponding events for all the essential state variables that are possible to be changed during runtime.

Alleviation

The MeerDefi team acknowledged this finding.

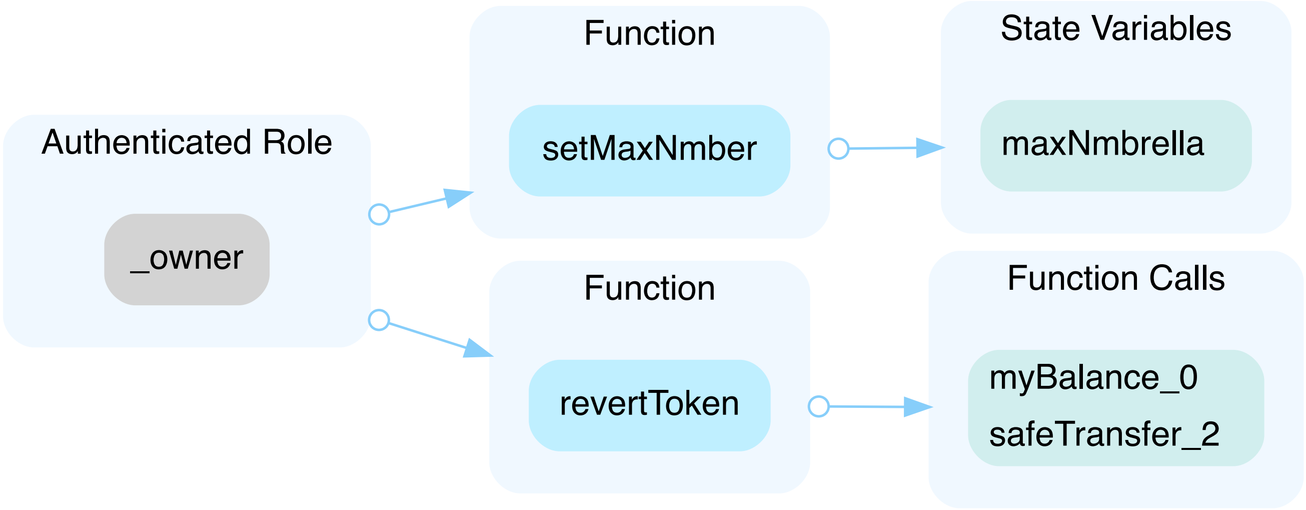
RCK-01 | Centralization Risk In Relation.sol

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Relation.sol: 27~29, 80~82	ⓘ Acknowledged

Description

In the contract `Relation` 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 MeerDefi team acknowledged this finding.

SLP-01 | Centralization Risk In StakeLP.sol

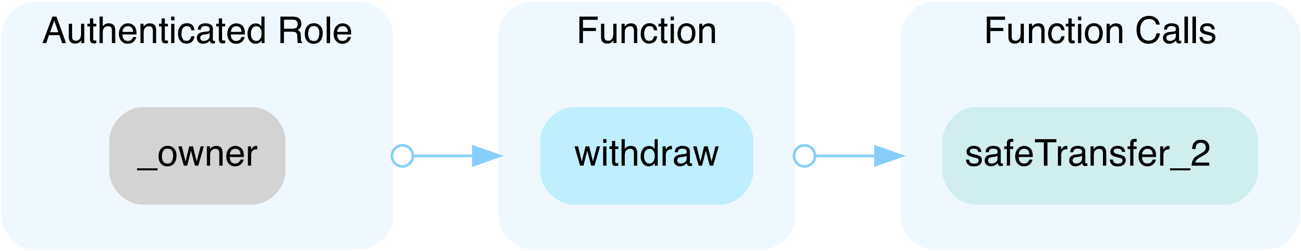
Category	Severity	Location	Status
Centralization / Privilege	● Major	stake/StakeLP.sol: 22~24, 31~33, 167~173, 175~178, 181~183, 186~188	ⓘ Acknowledged

Description

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

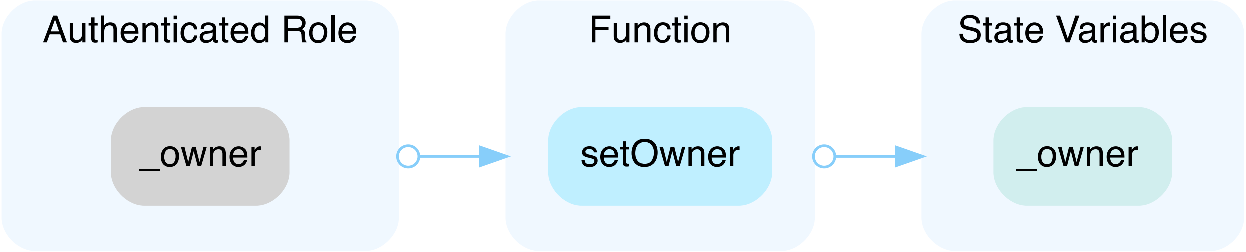
These `withdraw()` methods can extract the assets of these vault contracts.

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



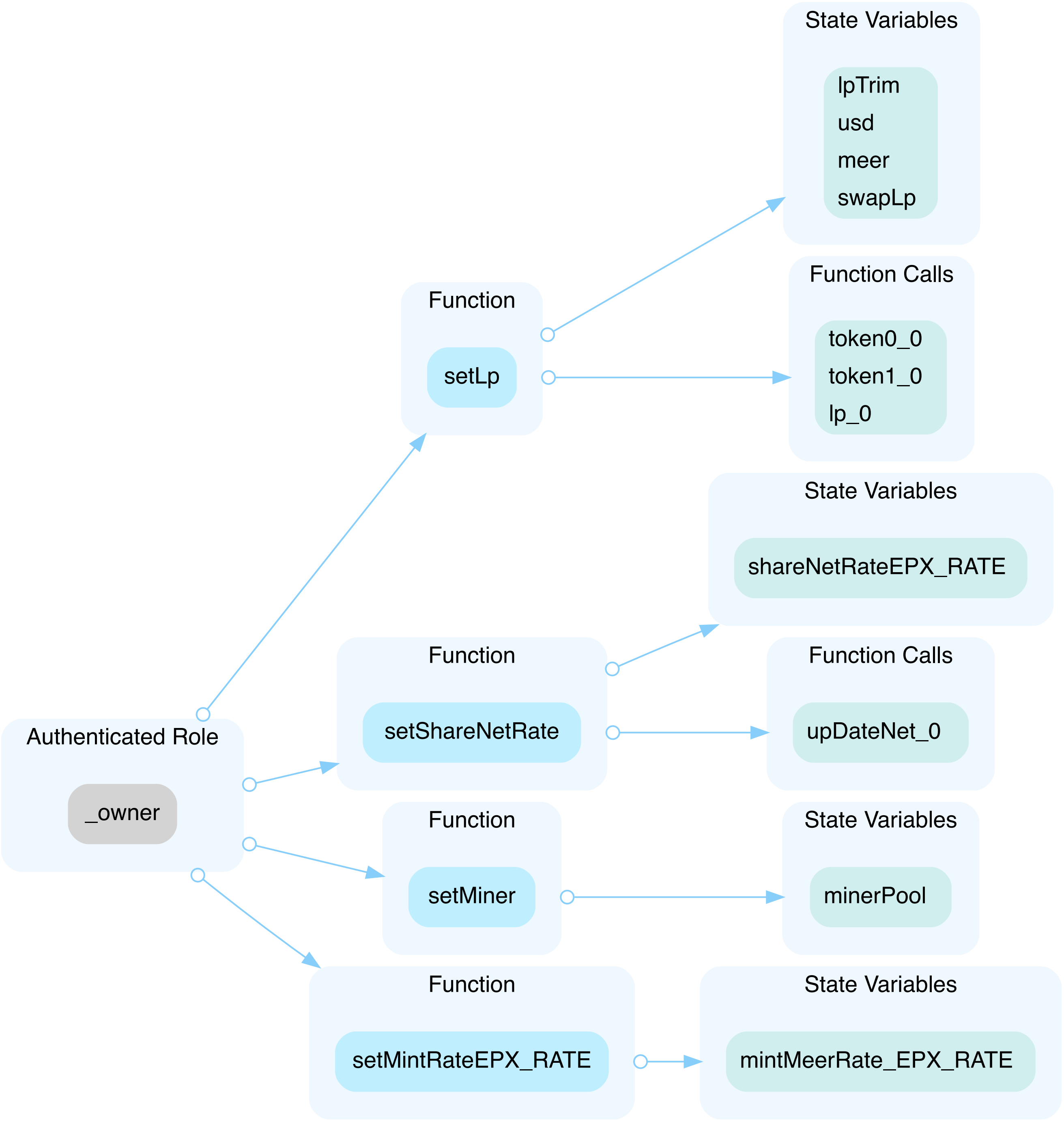
In the contract `Owner` the role `_owner` has authority over the functions shown in the diagram below.

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



In the contract `StakeMeerLp` 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., multi-signature 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 MeerDefi team acknowledged this finding.

SLP-02 | Lack Of Asset Transfer At UnStake() Function

Category	Severity	Location	Status
Logical Issue	● Major	stake/StakeLP.sol: 510	✓ Resolved

Description

When the `stake()` function is called, the user deposits USDT and records the amount. However, when `unStake()` is called, there is no logic or code to return the USDT to the user.

Recommendation

When the user redeems, the current contract shall return the deposited USDT to the user.

Alleviation

[MeerDefi]: This protocol will transfer USDT tokens to the caller in the function `lpTrim.removeLp()`.

[CertiK]: The implementation of `lpTrim.removeLp()` is out of scope for the current audit.

SLP-03 | Missing Check For Unlock Time

Category	Severity	Location	Status
Logical Issue	● Minor	stake/StakeLP.sol: 433	ⓘ Acknowledged

Description

When users call the `_withdrawLpAllFor()` function, it is necessary to check if the current time matches the unlock time. The code for current verification is commented out.

Recommendation

We recommend uncommenting to match the design intent.

Alleviation

The MeerDefi team acknowledged this finding.

SLP-04 | Lack Of Access Control On `claimFor()`

Category	Severity	Location	Status
Logical Issue	● Minor	stake/StakeLP.sol: 622	ⓘ Acknowledged

Description

Without validating the caller of the `claimFor()` method, anyone can call the `claimFor()` method and specify the `_owner_` address. This may cause users to be forced to withdraw their own rewards.

Recommendation

We recommend adding an access control for the caller.

Alleviation

The MeerDefi team acknowledged this finding.

SLP-05 | Unused Return Value

Category	Severity	Location	Status
Volatile Code	● Informational	stake/StakeLP.sol: 614	ⓘ Acknowledged

Description

The return value of an external call is not stored in a local or state variable.

File: projects/stake/StakeLP.sol (Line 614, Function `StakeMeerLp._mining`)

```
        IMiner(minerPool).mining();
```

Recommendation

We recommend checking or using the return values of all external function calls.

Alleviation

The MeerDefi team acknowledged this finding.

SLP-06 | Typo And Incorrect Comments

Category	Severity	Location	Status
Coding Style	● Informational	stake/StakeLP.sol: 548, 622, 626, 632, 633, 637, 637, 638, 639	ⓘ Acknowledged

Description

The name of the variable `rewrods` has a spelling mistake, it should be `rewards`.

The name of the function `upDateNet()` has a case error, it should be `updateNets()`.

The comment of the `stake()` function is incorrect. It should be `添加质押` instead of `赎回`.

The comment of the `unstake()` function is incorrect. It should be `赎回` instead of `添加质押`.

Recommendation

Please correct the spelling mistake.

Alleviation

The MeerDefi team acknowledged this finding.

SLP-07 | Number Of Denominator EXP_RATE

Category	Severity	Location	Status
Logical Issue	● Informational	stake/StakeLP.sol: 340~341	ⓘ Acknowledged

Description

When calculating awards for deposits, there are two `EPX_RATE`s in the denominator. When calculating the award for a stock, there is only one `EPX_RATE` in the denominator.

Is this because the reward multiplier for deposits is the product of a time multiplier and a quantity multiplier? The code logic should match the design intent

Recommendation

Financial models of blockchain protocols need to be resilient to attacks. They need to pass simulations and verifications to guarantee the security of the overall protocol.

The financial model of this protocol is not in the scope of this audit.

Alleviation

The MeerDefi team acknowledged this finding.

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.

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.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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