Midas

Smart Contract Security Audit

March, 2021

TechRate

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by Midas to perform an audit of smart contracts:

- ***https://github.com/MidasCore/midasgold-protocol/blob/b57626d6270c2f22bdbf65a7328911aeabf267f1/contracts/LayerRewardPool.sol***
- ***https://github.com/MidasCore/midasgold-protocol/blob/b57626d6270c2f22bdbf65a7328911aeabf267f1/contracts/MdgRewardPool.sol***

The purpose of the audit was to achieve the following:

- **Ensure that the smart contract functions as intended.**
- **Identify potential security issues with the smart contract.**

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Issues Checking Status

No.	Issue description.	Checking status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed

19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues and Owner privileges

1. Different total supply calculations

Issue:

In the LayerRewardPool contract total supply of mdg token calculated as sum of its total supply and burned amount.
In function with the same name in MdgRewardPool contract total supply of mdg token is calculated only as total supply.

Recommendation:

Please check this functions for correct evaluations of mdg token's total supply.

2. Block gas limit

Issue:

The `updateEmissionRate` function can fail due to block gas limit if the pool size is too big.

3. Zero address checking

Issue:

There is no zero address checking in functions initialize and initialize.

Recommendation:

Add zero address checking in these functions.

4. Double check

Issue:

There are double checks for _depositFeeBP is greater than some number in functions add and set.

Recommendation:

Please leave only the actual require message.

5. Hardcoded value

Issue:

There is a hardcoded pool id in function harvestAndRestake.

Recommendation:

Please leave the comment in code, why this value is equal to 8 or change it.

6. Wrong checking

Issue:

There is a wrong checking in add function for isStarted variable meaning.

Recommendation:

For checking that started you should check that the last reward block is greater than the start block.

```
bool _isStarted = (_lastRewardBlock >= startBlock) || (_lastRewardBlock <=
block.number);
```

7. Operator privileges

In the LayerRewardPool contract:

- ❑ Operator can change reserveFund.
- ❑ Operator can change the migrator.
- ❑ Operator can change the rates.
- ❑ Operator can change the reward per block.
- ❑ Operator can change the start block.

In the MdgRewardPool contract:

- ❑ Operator can change reserveFund.
- ❑ Operator can change the migrator.
- ❑ Operator can change the reward per block.
- ❑ Operator can change the mdo reward per block.
- ❑ Operator can change the bcash reward per block.

Conclusion

Smart contracts do not contain any high severity issues!

Techrate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.