Smart contract security audit

DEXMEX

v.1.0

Table of Contents

1.0 Introduction

1.1 Project engagement

During March of 2021, Dexmex engaged CTDSec to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. Dexmex provided CTDSec with access to their code repository and whitepaper.

Dexmex is a DEX based on uniswap that offers p2p leveraged trading.

1.2 Disclaimer

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the network's fast-paced and rapidly changing environment, we at CTDSec recommend that Dexmex team put in place a bug bounty program to encourage further and active analysis of the smart contract.

2.0 Coverage

2.1 Target Code and Revision

For this audit, we performed research, investigation, and review of the Dexmex contract followed by issue reporting, along with mitigation and remediation instructions outlined in this report. The following code files are considered in-scope for the review:

Source:

https://etherscan.io/address/0x0020d80229877b495d2bf3269a4c13f6f1e1b9d3#code

https://etherscan.io/address/0x5F541f7fCa689fD0c5C2D6656bDD6A7d84a738B9

https://etherscan.io/address/0x2D615c4e1ef8b6CF2a9c9d1157Ad9b616F4345EA#code

2.2 Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

№	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
22	Overpowered functions / Owner privileges	FAILED

3.0 Security Issues

3.1 High severity issues [0]

No high severity issues found.

3.2 Medium severity issues [2]

1. Lock privileges

Issue:
If the Dexmex contract is locked, owner and whitelisted accounts can transfer funds to any account.

Recommendation:
We recommend disallowing everyone to transfer funds when the contract is locked, or checking that sender and receiver of the funds are both whitelisted.

2. Owner privileges

Issue:
Owner can change the fees in the MasterMex contract to any number.

In order to solve both problems we encourage the project team to renounce ownership and share TX with the community.

3.3 Low severity issues [0]

No low severity issues found.

4.0 Summary of the audit

In order to solve both medium issues we encourage the project team to renounce ownership (Token and predictions contracts) and share TX with the community to solve overpowered functions.