Jorge Rodriguez

CYBERSEC CONTRACT AUDIT REPORT

Dotbase - CTDSEC.com

Introduction

During January of 2021, Dotbase 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. Dotbase provided CTDSec with access to their code repository and whitepaper.

Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bugfree status. The audit documentation is for discussion purposes only.

I always recommend having a bug bounty program opened to detect future bugs.

Coverage

Target Code and Revision

For this audit, we performed research, investigation, and review of the Dotbase 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:

- Dotbasefarming.sol

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.

Correctness of the protocol implementation [Result OK]

User funds are secure on the blockchain and cannot be transferred without user permission [Result OK]

Vulnerabilities within each component as well as secure interaction between the network components [Result OK]

Correctly passing requests to the network core [Result OK]

Data privacy, data leaking, and information integrity [Result OK]

Susceptible to reentrancy attack [Result OK]

Key management implementation: secure private key storage and proper management of encryption and signing keys [Result OK]

Handling large volumes of network traffic [Result OK]

Resistance to DDoS and similar attacks [RESULT KO]

Aligning incentives with the rest of the network [Result OK]

Any attack that impacts funds, such as draining or manipulating of funds [Result KO]

Mismanagement of funds via transactions [Result KO]

Inappropriate permissions and excess authority [Result OK]

Special token issuance model [Result OK]

Vulnerabilities

HIGH	MEDIUM	LOW
1	**1**	**1**

ISSUES

HIGH

Overpowered Function

The owner of the contract has absolute power to drain all tokens if he sees fit.

This function is not trusted so it must be limited for example by adding an IF that determines that the tokens are not any of the current pools available.

Location:

MEDIUM SWC-128

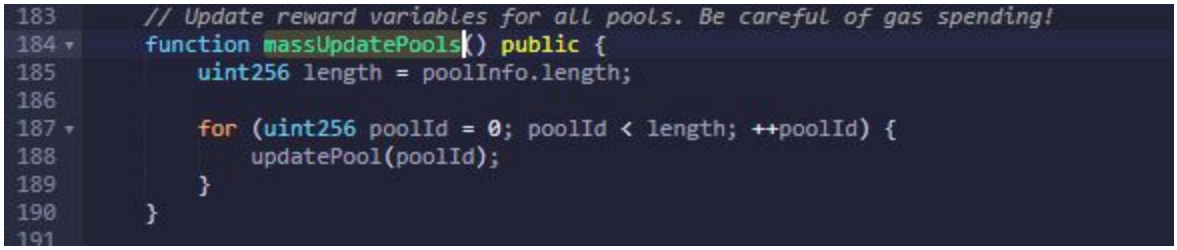
Loop over unbounded data structure.

Gas consumption in function "massUpdatePools" in contract "DotbaseFarming" depends on the size of data structures or values that may grow unboundedly. If the data structure grows too large, the gas required to execute the code will exceed the block gas limit, effectively causing a denial-of-service condition. Consider that an attacker might attempt to cause this condition on purpose.

Caution is advised when you expect to have large arrays that grow over time. Actions that require looping across the entire data structure should be avoided.

If you absolutely must loop over an array of unknown size, then you should plan for it to potentially take multiple blocks, and therefore require multiple transactions.

Locations



LOW - SWC 110

An assertion violation was triggered.

It is possible to cause an assertion violation. Note that Solidity assert() statements should only be used to check invariants. Review the transaction trace generated for this issue and either make sure your program logic is correct, or use require() instead of assert() if your goal is to constrain user inputs or enforce preconditions. Remember to validate inputs from both callers (for instance, via passed arguments) and callees (for instance, via return values).

Locations

Summary of the Audit

The team **must mitigate and fix the HIGH issue** presented in the report so that the contract can be deployed, also we recommend to review the medium/low ones.
All the logic functions presented by the team have been validated with test cases and fulfill their function correctly, we will deliver the test cases to the team.