Smart contract security audit

AwooFinance

v.1.2

Table of Contents

1.0 Introduction

1.1 Project engagement

During June of 2021, AwooFinance 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. AwooFinance provided CTDSec with access to their code repository and whitepaper.

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 AwooFinance 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 AwooFinance 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:

AwooFinance.sol [SHA256] -

694a87a0b05dc828d845ae97e6c16b38d12b4f5d4cdd0d77d26500d5c98c2093

DogNFT.sol [SHA256] - f1631a8f74002f5e5ecf2de4fc991639c19a3f112c51a8a65d2f01733bea1dc1

TreatsNFT.sol [SHA256] - 2f1cac4996c1db89bdc24b68f9a2363651ac89aa21b8ade890223a4c16bf99d9

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.	LOW ISSUES
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.	SOLVED BY DEV TEAM
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	PASSED

3.0 Security Issues

3.1 High severity issues [1] - Solved

1. Economy error Issue:

The function _beforeTokenTransfer() in DogNFT contract pushes addresses to the boosters list. But do not increase _totalBoosters counter.

That will cause calculation errors in the calculatedSupply() function.

Recommendation:

Increment _totalBoosters variable.

Dev team update: Solved by dev team in commit 01411f772becfa62cf45493907386e82c96464be.

3.2 Medium severity issues [0]

No medium severity issues found.

3.3 Low severity issues [2]

1. Out of gas Issue:

The function _beforeTokenTransfer() in DogNFT contract uses the loop to find an empty account. Function will be aborted with OUT_OF_GAS exception if there will be a long boosters list.

The function calculatedSupply() in the AwooFinance contract uses the loop to calculate supply. It also could be aborted with OUT_OF_GAS exception if there will be a long boosters list.

Recommendation:

Use enumerable sets instead of arrays.

2. Optimization Issue:

The function _transfer() in AwooFinance contract has part with executing the same lines of code as manualSend() function.

Recommendation:

Make manualSend() function public and use it inside _transfer() function.

4.0 Owner Privileges

AwooFinance:

Owner can exclude the address from the fee.

Owner can change taxFee, charityFee and opFee in valid range.

Owner can change charity and op wallets.

Owner can change the max transaction amount.

Owner can manually swap and send ETH to charity and op wallets.

Owner can enable and disable autoswap on transactions.

DogNFT:

• Owner can mint.

• Owner can change:

1. treats NFT
2. base uri

TreatsNFT:

Owner can change token uri.

Owner can mint.

5.0 Summary of the audit

Smart contracts contain low issues and it’s safe to deploy.