www.theblockchainauditor.com

TWO TWO Smart Contracts Audit

Prepared by:
Jorge Martinez
Walther Villatoro

Version: 3
Date: Aug 6, 2021

Table of Contents

1. Project Information 3

1.1 Project Scope 3

1.2 Issue Classification 4

2. Findings 5

3.1 Breakdown 5

3.2 Test Results 6

3. Detailed Results 7

3.1 Differences between ERC20 & BEP20 7

1. Project Information

1.1 Project Scope

We were tasked with auditing the Two Two Extreme Art smart contract platform.GIT This audit process pertains to a github repository provided by the Two Two Extreme Art development team on July 30th, 2021. The files within scope of this audit are:

File	MD5
./contracts/X22.sol	22e113d61bbc4deddb46f6468eddded2
./contracts/StakingRewards.sol	0de5c2745ad509c2f7a7000fa8a2114e
./contracts/LiquidityProvision.sol	bc70a57a40b676a3945f1bcd828c1c0e

And can be found at here https://github.com/TWOTWOArt/TwoTwoArt-contracts on git commit 749c6b8028532ad0864e3e26e0d9984cb17898a1.

X22.sol is an ERC20 & BEP20 compliant token contract and StakingRewards.sol and LiquidityProvision.sol is the corresponding staking contract meant to power the Two Two Extreme Art platform.

3.2 Test Results

```
Two Two Art Security Liquidity Provision Test Suite
  LP Staking functionality
    ✓ A user should be able to stake LP tokens into the contract (60ms)
    ✓ A user cannot stake zero LP tokens into the contract
    ✓ Get reward after some time (75ms)
    ✓ A user that stakes for less than 8 days pays a 7% fee (52ms)
    ✓ A user must manually begin the withdrawal procedure and input false to not pay fees (43ms)
    ✓ A user that starts the withdrawal process before claiming won't pay fees if they waited 8 days (55ms)
    ✓ users that claim before 8 days can choose to pay the fee to withdraw tokens (61ms)
    ✓ A user who claims before the withdrawal procedure is done will have their transaction reverted (59ms)
    ✓ A user that is staking cannot withdraw 0 tokens
    ✓ A user can use the exit function to quickly get rewards and funds minus a 7% fee (83ms)
    ✓ A staker can begin a withdrawal and then cancel it at any moment to keep staking (55ms)
    ✓ The owner of the project can change the wallet the fees go to
    ✓ The reward rate is based on the rewards duration
    ✓ If the owner wants to add more rewards, the notifyRewardAmount will adjust the rewardRate (44ms)
    ✓ Staked rewards are reserved for stakers (100ms)
    ✓ should revert if the contract doesn't have reward tokens (46ms)

Two Two Art Security Staking Rewards Test Suite
  Staking functionality
    ✓ A user should be able to stake X22 tokens into the contract
    ✓ A user cannot stake zero X22 tokens into the contract
    ✓ Get reward after some time (53ms)
    ✓ A user that stakes for less than 8 days pays a 7% fee (57ms)
    ✓ a user must manually begin the withdrawal procedur to not pay fees (55ms)
    ✓ A user that starts the withdrawal process before withdrawing won't pay fees if they waited 8 days (81ms)
    ✓ users that claim before 8 days can choose to pay the fee to withdraw tokens (58ms)
    ✓ IMPORTANT: a user who claims before the withdrawal procedure is done will have their transaction reverted (53ms)
    ✓ a user that is staking cannot withdraw 0 tokens
    ✓ a user can use the exit function to quickly get rewards and funds minus a 7% fee (78ms)
    ✓ A staker can begin a withdrawal and then cancel it at any moment to keep staking (50ms)
    ✓ A user that has earned rewards from staking can stake those rewards (62ms)
    ✓ the owner of the project can change the wallet the fees go to
    ✓ the reward rate is based on the rewards duration
    ✓ if the owner wants to add more rewards, the notifyRewardAmount will adjust the rewardRate (95ms)
    ✓ staked rewards are reserved for stakers (101ms)

Two Two Art Security X22 Token Test Suite
  Deployment
    ✓ Name should be X22
    ✓ Symbol should be X22
    ✓ Should have 18 decimals
    ✓ Should have a total supply of two million  tokens
    ✓ Should send the total supply to the address that deploys the contracts
  Transfer function behavior
    ✓ User should be able to transfer tokens
    ✓ A third party can transfer tokens from one user to another
    ✓ Should decrease allowance
  Testing ownership
    ✓ renounce ownership sets zero address as owner
    ✓ only the owner can transfer ownership
  Minting
    1) The minter is the owner of the contract
    2) the owner can change the minter even after changing the minter
    3) only the minter can mint tokens
    4) cannot mint to the zero address
    5) cannot mint more than 22 million and 2 tokens
  Burning
    6) everybody can call the burn function
    7) nobody can burn more tokens than they own
  Whitelist permissions
    ✓ Only whitelister address should be able to transfer whitelister permission
    ✓ cannot set whitelister to zero address using transferWhitelister, must use renounceWhitelister
    ✓ Only whitelister address should be able to transfer Whitelister permission
    ✓ Only the whitelister address should be able to renounce Whitelister permission
  Whitelist behavior
    ✓ whitelister should be the caller
    ✓ Should revert createLODWhitelist if duration and amountMax array lengths are not equal
    ✓ Should revert modifyLODWhitelist if index is greater than  lgeWhitelistRounds length
    ✓ Indexed duration and amountMax are updated if different values are sent in modifyLODWhitelist
```

1.2 Issue Classification

Informational

This issue relates to style and security best practices but does not pose an immediate risk.

Low

An issue classified as informational does not pose an immediate threat to disruption of functionality and could not be exploited on a recurring basis, however, it should be considered for security best practices or code integrity.

Medium

An issue classified as medium has relatively small risk and isn't exploitable to circumvent desired functionality and could not have financial consequences but could put user's sensitive information at risk.

Critical

These issues in the smart contract can have catastrophic implications that could ruin your reputation, disrupt the contract's functionality, and impact the client and your user's sensitive information.

2. Findings

3.1 Breakdown

We were tasked with auditing their liquidity pool token staking contract named LiquidityProvision.sol along with doing a regression analysis verifying that the new changes did not distrub existing functionality.

LiquidityProvision.sol and StakingRewards.sol were both based on the Synthetix StakingRewards.sol that was modified and integrated into the X22 ecosystem. This made it possible to reference existing audits in the analysis gave us assurance in focusing on the modificatoins and the X22 and LP token integration into the Two Two Art smart contract platform.

0	0	0	1
Critical Issues	Medium Issues	Low Issues	Informational Issues

We did notice that there were some slight differences between the ERC20 version of X22.sol and the BEP20 version, which are highlighted in our test suite. The BEP20 will have mint(), burn(), functions along with a minter role while the ERC20 version will not. Besides the slight differences no actual issues were found in the X22 Token.

The X22 staking contract, StakingRewards.sol, passed all of the regression tests.

A testing suite was written for LiquidityProvision.sol and no issues were found within that contract and were able to verify the LP staking functionality. https://github.com/ethereumbook/ethereumbook/blob/develop/09smart-contracts-security.asciidoc

3. Detailed Results

Informational Issues

3.1 Differences between ERC20 & BEP20

The BEP20 that was pushed on this commit doesn't have the mint() and burn() functions whereas previously the ERC20 did. I was told this was by design so its simply an informational issue.

Working with Two Two Extreme Art was very enjoyable. In this audit, we only found 1 Informational Issue, which had no impact on the expected functionality of the platform. With the smart contracts now being mainnet ready, we are looking forward to seeing these contracts in production and are proud to have been a part of making the Two Two Extreme Art smart contract platform safer and more reliable.

The Blockchain Auditor