MasterApeAdmin

smart contracts
final audit report

November 2021

hashex.org

contact@hashex.org

Contents

1. Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice 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 disclaimer below – please make sure to read it in full.

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 HashEx and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (HashEx) owe no duty of care towards you or any other person, nor does HashEx 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 HashEx 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, HashEx hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HashEx, 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 the 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. HashEx owns all copyright rights to the text, images, photographs, and other content provided in the following document. When using or sharing partly or in full, third parties must provide a direct link to the original document mentioning the author (hashex.org).

2. Overview

HashEx was commissioned by the ApeSwap Finance team to perform an audit of their smart contracts. The audit was conducted between October 31 and November 02, 2021.

The code located in the GitHub repository @apeswapfinance/apeswap-banana-farm was audited after the commit 3ea6a0c. The MasterApeAdmin contract is proxy owner for the ApeSwap MasterApe. Documentation was provided with the in-code comments and NatSpec descriptions. The repository contains tests for the reviewed contract.

The purpose of this audit was to achieve the following:

- Identify potential security issues with smart contracts.
- Formally check the logic behind given smart contracts.

Information in this report should be used to understand the risk exposure of smart contracts, and as a guide to improving the security posture of smart contracts by remediating the issues that were identified.

Update: the recheck was done after commit 9d89982. The audited code was published to the BSC network at address 0x9fed2bc7f0b4f4350b52e29e0a3c2bf5ebc3cc0a.

2.1 Summary

Project name	MasterApeAdmin
URL	https://apeswap.finance
Platform	Binance Smart Chain
Language	Solidity

2.2 Contracts

Name	Address
MasterApeAdmin	0x9FEd2bC7F0b4F4350B52E29e0a3c2Bf5EbC3cc0A

3. Found issues

High	1 (20%)
Low	2 (40%)
Info	2 (40%)

C1. MasterApeAdmin

ID	Severity	Title	Status
C1-01	High	Unlimited emission rate	Resolved
C1-02	Low	Multiple reads in for() loops	Acknowledged
C1-03	Low	Gas optimization of fixedPercentFarmPids[]	Acknowledged
C1-04	Info	Unused variable FixedPercentFarmInfo.pid	Acknowledged
C1-05	Info	Immutable variable masterApe	Resolved

4. Contracts

C1. MasterApeAdmin

Overview

Admin MasterApe proxy contract used to add features to MasterApe admin functions.

Issues

C1-01 Unlimited emission rate — High — Resolved

`updateMasterApeMultiplier()` function L98 should require the `_newMultiplier` value to have an upper limit. Setting the unlimited emission rate in the MasterApe would cause an instant BANANA price crash in case of losing control over the MasterApeAdmin's owner account.

Recommendation

Implement the `require()` check for the updated multiplier value.

C1-02 Multiple reads in for() loops — Low — Acknowledged

Multiple reads of the same variables over for() loops could be avoided using the local variables, see L284, 332, 342, 344.

C1-03 Gas optimization of fixedPercentFarmPids[] — Low — Acknowledged

EnumerableSet could be used instead of `fixedPercentFarmPids[]` + `getFixedPercentFarmFromPid[]`. Another option is refactoring this pair into `fixedPercentFarmPidToIndex[]` mapping + `getFixedPercentFarm[]` array. Both solutions save gas, removing the for() loop L283.

C1-04 Unused variable FixedPercentFarmInfo.pid ● Info Acknowledged

No need to store `pid` parameter in the `pid=>FixedPercentFarmInfo` mapping L43.

C1-05 Immutable variable masterApe ● Info Resolved

`masterApe` address could be declared immutable in order to save gas. The admin proxy contract doesn't imply future upgrades in its current form.

5. Conclusion

1 high severity issue was found and resolved with the code update. The contracts are highly dependent on the owner’s account. Users using the project have to trust the owner and that the owner's account is properly secured.

This audit includes recommendations on the code improving and preventing potential attacks.

Appendix A. Issues' severity classification

- **Critical.** Issues that may cause an unlimited loss of funds or entirely break the contract workflow. Malicious code (including malicious modification of libraries) is also treated as a critical severity issue. These issues must be fixed before deployments or fixed in already running projects as soon as possible.
- **High.** Issues that may lead to a limited loss of funds, break interaction with users, or other contracts under specific conditions. Also, issues in a smart contract, that allow a privileged account the ability to steal or block other users' funds.
- **Medium.** Issues that do not lead to a loss of funds directly, but break the contract logic. May lead to failures in contracts operation.
- **Low.** Issues that are of a non-optimal code character, for instance, gas optimization tips, unused variables, errors in messages.
- **Informational.** Issues that do not impact the contract operation. Usually, informational severity issues are related to code best practices, e.g. style guide.

Appendix B. List of examined issue types

- Business logic overview
- Functionality checks
- Following best practices
- Access control and authorization
- Reentrancy attacks
- Front-run attacks
- DoS with (unexpected) revert
- DoS with block gas limit
- Transaction-ordering dependence
- ERC/BEP and other standards violation
- Unchecked math
- Implicit visibility levels
- Excessive gas usage
- Timestamp dependence
- Forcibly sending ether to a contract
- Weak sources of randomness
- Shadowing state variables
- Usage of deprecated code

contact@hashex.org

@hashexbot

blog.hashex.org

linkedin

github

twitter

HashEx

Blockchain Security