0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

BaseBOMB

September 2023

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for **BaseBOMB**.

The audited BaseBOMB project is a Pancakeswap fork with UniswapV2-like DEX and Sushiswap-like Masterchef farming.

Contracts were provided in .sol files with following SHA-1 hashes:

MasterChef.sol ee2b59fac05e96e8360f06defd4ac29d7e1e6529

BnbStaking.sol bf9b950744c5d07c42f58fc2a6ac415f04b0c550

BombToken.sol 418aa3c15fbda57eb3c64a03a2796c1f0ef8d685

OpBombRouter.sol 08595d996f6b214cf0aa1a8d9e26ffa702fa03b5

OpBombRouter01.sol 278248ba12199eb421394b542c8a16579f286d92

IERC20.sol 0adf28453394a949734befc968a2f2e79c40338d

IOpBombERC20.sol 6634e76755755cf016530ee6dda99a92ee62b46c

IOpBombRouter01.sol b6ec1022fa35022f6238577a5ea79344b37bd220

IOpBombFactory.sol ee486a45e32394695b4cae5491ee2dec5119eb6d

WBNB.sol a070a2f6b1b93b6c68d38f91748bff07218a3119

IOpBombCallee.sol fb5cdb4a68e13a0f996e57ee3d98d1e62e2a250b

IOpBombPair.sol 4c78d9162e4a20f49a106e8e7033b4bc524a4b56

IWETH.sol ffbbf309951e053bcb23c2b55815f7c0d605f907

IOpBombRouter02.sol fe52944ff88df83b7e0790d66852c6c16b13b601

CoinFactory.sol 84a855335cc55e0833086b01aeab8c993f9c65b7

OpBombPair.sol f1ce7dae6322f959d869264b14160f8b9fe12a97

OpBombFactory.sol e2dbe892c43513be71b063a3380c5c54ff4abaa8

OpBombPresale.sol cd7493429ac18888eea8d8d778df4af106903063

OpBombERC20.sol a456c3b1b370ea9aef109d94564091881af162dc

TokenTimelock.sol ebd5dbd664cc63fc53934a1d4442b1d4c4e237c1

UQ112x112.sol 4fa73e546faecb76d3ca7966ee2149967e613dc6

Math.sol ba812752e4ed21669dc5122320a907c61ba31119

SafeMath.sol 9aa9d954ceb67415a9ba06c3cbf82c87996315c0

Babylonian.sol b76ad4b7ba11fcb6c20e4e283268387f729ccbd8

WETH.sol 63aecd6655684d1b09c76d76c8be0d45d177d595

OpBombLibrary.sol 92925c9a8695dd8645c68753b54d946f36f31652

Update. The updated code was deployed to the following addresses in the Base Chain:

BombToken 0x2774eF7918F6947560FD7128878Cfee3F34535e6

BaseBombFactory 0xBAa207A1673dea8b6890817e5e68e06677471CFB

BaseBombRouter 0x411f222a24EB69654341303D09c0b0bD87aCf7Fa

MasterChef 0xE5A2ec9982bb890F633B5451da229683f5A7b51a

Name	BaseBOMB
Audit date	2023-08-24 - 2023-09-10
Language	Solidity
Platform	Base Chain

Contracts checked

Name	Address
OpBombPair	
BombToken	
MasterChef	

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed
Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed

Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Delegates minting (BombToken)
Status: Fixed

The checkpoint history balances can't be used since delegates can be minted. The `_moveDelegates` function must be used with every transfer but in fact it's called only with the `mint` function.

```
// BombToken with Governance.

contract BombToken is ERC20("Bomb Token", "Bomb"), Ownable {
    function mint(address _to, uint256 _amount) public onlyOwner {
        _mint(_to, _amount);
        _moveDelegates(address(0), _delegates[_to], _amount);
    }
...
}
```

Recommendation: Fix transfers or remove checkpoint functionality to reduce gas consumption if governance with the BombToken is not needed.

2. Syrup is not burnt (MasterChef)

Status: Fixed

The original Syrup workflow is to be minted for the users who have entered the first pool of Masterchef and to be burned when user is withdrawing. But in fact, Syrup token is not burned in the `emergencyWithdraw` function resulting in open mint of syrup tokens using `deposit - emergencyWithdraw -...` scheme.

Recommendation: Burn Syrup tokens in `emergencyWithdraw` function or disable it completely.

Medium severity issues

1. Syrup minting (MasterChef)

Status: Fixed

The `enterStaking` function is a deposit function for a particular pool with `pid=0`. The original Pancakeswap Syrup is minted to user, but BaseBOMB Masterchef mint syrup to the `feeAddress` and only if the first pool's deposit fee is non-zero. Syrup token in the current code is useless as it's minted only for the project owner.

```
function enterStaking(uint256 _amount) public {
    if (pool.depositFeeBP > 0) {
        uint256 depositFee = _amount.mul(pool.depositFeeBP).div(10000);
        syrup.mint(feeAddress, _amount);
        user.amount = user.amount.add(_amount).sub(depositFee);
    } else {
        user.amount = user.amount.add(_amount);
...
}
```

Recommendation: Fix the code or add documentation.

Low severity issues

1. Incorrect comments (OpBombPair)

Status: Open

Incorrect comments in OpBombPair:

```
uint public constant MINIMUM_LIQUIDITY = 10**2; // 10 ** 3

uint public constant MIN_FEE_AMOUNT = 0; // = 0.01%

uint public constant MAX_LP_FEE_AMOUNT = 25; // = 1%

uint public constant MIN_LP_FEE_AMOUNT = 0; // = 0.01%
```

2. Some tokens aren't supported (MasterChef)

Status: Open

Tokens with fees on transfers or tokens with transfer hooks are not supported as staking tokens. The owner must not add pools with such tokens to the Mastrechef.

Conclusion

BaseBOMB OpBombPair, BombToken, MasterChef contracts were audited. 2 high, 1 medium, 2 low severity issues were found.
2 high, 1 medium severity issues have been fixed in the update.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

0x Guard