Audit report for Bumper Finance
14/07/2021
v.1.0

Bumper Finance token audit report

1. Document version

Version	Date	Author	Description
1.0	14.07.2021	Tomasz Kęczkowski	Final version
0.1	07.07.2021	Tomasz Kęczkowski	Initial draft, findings added

2. Executive summary

The following audit report presents the effect of the research that Blockhunters team conducted on the part of the Bumper Finance Tokens code. The research was made from 06/22/2021 till 07/14/2021 on the code delivered by Block8 team on their Gitlab dev branch.

Our audit focused on six contracts of the project – BumpToken, BumpMarket, BUSDC, DataTimeUtils, GovernanceAndOwner, TimeLockMechanism.

Blockhunters team has checked the possibility of known Ethereum attacks to be exploited on the contracts. Fortunately, the main token contracts contain basic functionalities that are not vulnerable to discovered Ethereum attacks. BumperToken developers has been using SafeMath libraries that significantly lower the risk of possible miscalculations and errors. All of the contracts, methods and state variables were tested and stated as safe to use.

The smart contracts were created in a secure way, no major bugs were found by our team. We are happy to inform that the code that our team has tested is free of potential critical errors and can be safely used in the network.

All of suggestions and recommendations were applied to the code after suggestions delivered in the first version of the report on 07/07/2021. Overall quality of the smart contract code and tests is very good.

VERIFIED
BLOCK HUNTERS
APPROVED

2.1. Liability clause

Please note that Blockhunters Company doesn’t verify the economic foundation of the project but only its code correctness and security issues. We do not take any responsibility for any misuse or misunderstanding of the information provided and potential economic losses due to faulty investment decisions. This document doesn’t ensure that the code itself is free from potential vulnerabilities that were not found. If any questions arise please contact us by www.blockhunters.io.

2.2. Commit hash and MD5 hashes

Before using the smart contracts, please verify MD5 commit hashes with the following ones, which describe the files that were audited between 24th of June and 14th July 2021.

Commit	6756c093813b6db826d25ae821f61e355c6a1305

Filename	MD5
/BUMPToken.sol	c445dbdaf5845b2c987ba2a0dd487180
/BUSDC.sol	cd179fe5d99b7857a2a2dbdf6016a0ed
/BumpMarket.sol	89b2f642eed239b087d425a3b51031e5
/BumperAccessControl.sol	179a1a0b8816c0763a0377b71b484457
/Migrations.sol	1fb5e319b84b376a1a83b52894ab7aea
/TimeLockMechanism.sol	dcec1d492ee4e262ccec4ab23b66a0c9
/interfaces/IVault.sol	2da439020dd78951bf9501968c17f784
/stubs/YearnStub.sol	40bc937c063e5537810dc185f7c296ba
/testContracts/BUMPTokenUpgrade.sol	0fb044c679ba1cfee08b28545555b714
/testContracts/BUSDCUpgrade.sol	5c2fe66a21cc93580d640a34c8af39c9
/testContracts/BumpMarketTest.sol	fb8e2f0ab032d8dae2c2124b66f3d30a
/testContracts/BumpMarketUpgrade.sol	12835fa6eb55fdf9f82204844b371480
/testContracts/BumpTokenTest.sol	52fc82fd479731ab2151457bda3b5bbc

2.3. Table of contents

3. Main contract audit

3.1. Errors known from Ethereum

✓ Reentrancy attack

Non-susceptible. The contracts adhere to ERC20 protocol and use OpenZeppelin standards where possible. Critical methods that manipulate funds are protected with nonReentrant modifier and are therefore safe against these types of attacks.

✓ Race conditions

Flow of the system is linear and straightforward. Nothing time-sensitive and requiring synchronicity is performed.

✓ Integer over / underflow

Contracts use the newest solc version where SafeMath library is built-in, which prevents this class of errors.

✓ Timestamps

Custom logic dependent on block.timestamp is a source of many leaks as it can be influenced by the miners. The contract is safe from any such attacks.

✓ Library dependencies

All used dependencies are in the source files.

✓ Front-running

Front running isn’t a risk for integrity of the system with it’s current capabilities. Foreseeing transactions before visible in the block won’t have any bad results for the users.

✓ DoS

Neither of the contracts can be rendered inoperable by the users

✓ Insufficient gas griefing

Non-susceptible. The contracts don’t use any low-level contract calls, thus this error won’t occur. This attack may be possible on a contract which accepts generic data and uses it to make a call another contract (a 'sub-call') via the low-level address.call() function, as is often the case with multi-signature and transaction relay contracts.

✓ Token deposit and creation

Asset flow follows the specification models and the logic is well tested for integration external smart contracts

3.2. Automated and manual audit

✓ Mythril

- Version number: v0.22.21
- Performed by: PP
- Checked by: RC
- Date, time: 2.07.2021
- Results: No vulnerability detected

✓ Slither

- Version number: 0.7.1
- Performed by: PP
- Checked by: RC
- Date, time: 1.07.2021
- Results: No vulnerability detected

✓ Tokenguard

- Version number: alpha
- Performed by: PP
- Checked by: RC
- Date, time: 5.07.2021
- Results: No vulnerability detected

4. Bumper Token overview / methods checked

4.1. BumpToken

Method	Status	Information
BUMPToken.distributeToAddress	OK	
BUMPToken.initialize	OK	
BUMPToken.mint	OK	
BUMPToken.updateUnlockTimestamp	OK	

4.2. BumpMarket

Method	Status	Information
BumpMarket.approveUSDCToYearnVault	OK	
BumpMarket.depositAmount	OK	
BumpMarket.depositUSDCInYearnVault	OK	
BumpMarket.estimateBumpRewards	OK	
BumpMarket.estimateSwapRateBumpUsdc	OK	
BumpMarket.getBumpAllocation	OK	

BumpMarket.getBumpPurchaseAmount	OK	
BumpMarket.getBumpRewards	OK	
BumpMarket.getCurrentPrice	OK	
BumpMarket.getyUSDCIssuedToReserve	OK	
BumpMarket.initialize	OK	
BumpMarket.updateBumpPurchaseAllocation	OK	
BumpMarket.updateBumpRewardAllocation	OK	
BumpMarket.updateBusdcUnLockTimestamp	OK	
BumpMarket.updateMaxBumpPercent	OK	
BumpMarket.withdrawUSDCFromYearnVault	OK	

4.3. BUSDC

Method	Status	Information
BUSDC.decimals	OK	
BUSDC.initialize	OK	
BUSDC.mint	OK	
BUSDC.pause	OK	

BUSDC.unpause	OK	
BUSDC.updateUnlockTimestamp	OK	

4.4. GovernanceAndOwner

Method	Status	Information
GovernanceAndOwner._GovernanceAndOwner_init	OK	
GovernanceAndOwner.addAddressInWhiteList	OK	
GovernanceAndOwner.onlyGovernance	OK	Note: onlyGovernance has been set to a Gnosis Multisig Wallet, located at 0x486DD7c8FEE800400615Fd7952E8e3b6071b2FF5 While the Gnosis Multisig is not a part of the audit, we note that it has been independently audited by G0 Group. The audit for version 1.1.1 can be found at https://github.com/gnosis/safe-contracts/blob/v1.1.1/docs/Gnosis_Safe_Audit_Report_1_1_1.pdf
GovernanceAndOwner.onlyGovernanceOrOwner	OK	

GovernanceAndOwner.removeAddressFromWhiteList	OK	

4.5. TimeLockMechanism

Method	Status	Information
TimeLockMechanism._TimeLockMechanism_init	OK	

5. Tests

We verified the test suite provided by Block8's crew. It has been written in an exhaustive and correct manner, thus guaranteeing the correct operation of the system. Coverage of crucial logic is high (~90%). Based on that we didn’t develop any additional tests.

6. Attachments

Slither & Mythril logs will be sent in a separate files.

Thank you!

Contact us at:

heyhunters@blockhunters.io

www.blockhunters.io