BLOCKCHAIN FOCUS GROUP

Smart Contract Security Audit

For ExoniumDEX

21 July 2021

About BCFG Security Audit

From prototype to production, BCFG Audit team works closely with projects through the complete development cycle to identify security issues and recommend best practices. Our mission is to mitigate and pinpoint all possible security risks and flaws within each line of code. All audit reports will be tailored to different clients, unlike some audit firms that provide identical reports to different projects.

Other than establishing a secure platform for our clients, we strive to promote a secured and safe environment for the DeFi community as well.

For audit enquiries, please reach out to hello@bcfg.io.

Website: https://bcfg.io

Table of Contents

Disclaimer

Blockchain Focus Group ("BCFG") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided by the paying client for the scope of this audit. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the smart contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

This report is not an investment advice, nor does it guarantee adequacy of a business model and/or a bug-free code. This report should be used only to discuss known technical problems. The code may include problems on Solidity that are not included in this report. It will be necessary to resolve addressed issues and conduct thorough tests to ensure the safety of the smart contract. Information is provided 'as is', and BCFG is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will BCFG or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report. BCFG does not recommend that any cryptocurrency should be bought, sold, or held by you. Blockchain technology and cryptocurrencies are highly volatile and speculative by nature. Do conduct your own due diligence and consult your financial advisor before making any investment decisions. users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence, or approval by BCFG.

1 Overview

This report has been prepared for ExoniumDEX. BCFG provides a user-centered examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	ExoniumDEX
Website	https://exonium.one/
Blockchain	Binance Smart Chain & Polygon
Language	Solidity
Relevant Links	Whitepaper: https://exonium.one/tEXO-Whitepaper.pdf Docs: https://texo.gitbook.io/exoniumdex Medium: https://medium.com/exonium-exchange Facebook: https://www.facebook.com/Exoniumdex LinkedIn: https://www.linkedin.com/company/exoniumdex Github: https://github.com/EXONIUM-DEX Twitter: https://twitter.com/ExoniumDex Telegram Group: https://t.me/exoniumofficial Telegram Channel: https://t.me/ExoniumDEX

1.2 Contract Assessed on BSC

Token	https://bscscan.com/address/ 0xF1afb5674Bf946458BD1163163F62dE683B07D65 **(TEXO)** https://bscscan.com/address/ 0xE2faB6C603C70aae2CCDEF140DCd8b7d6CCDFf35 **(FAANG)**
Masterchef	https://bscscan.com/address/ 0xD8980CCdD4096e60bb3198F91d6f79CeEF29369c **(TEXOOrchestrator)** https://bscscan.com/address/ 0xeBe1F3221FB51Ddab6d760Bda3fabb288c1b8EE7 **(FAANGOrchestrator)**
Timelock	https://bscscan.com/address/ 0xC52329f4164E115ca40B3D5Ab8E1FF3F85E9A253

1.3 Issues on BSC

	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High Severity	0	0	0	0
Medium Severity	0	0	0	0
Low Severity	0	0	0	0
Total	0	0	0	0

1.4 Contract Assessed on Polygon

Token	https://polygonscan.com/address/0xF1afb5674Bf946458BD1163163F62dE683B07D65 **(TEXO)** https://polygonscan.com/address/0xAa40849abB7Ba1689cC1272f0D00Ccc04c9bAA04 **(FAANG)**
Masterchef	https://polygonscan.com/address/0xD8980CCdD4096e60bb3198F91d6f79CeEF29369c **(TEXOOrchestrator)** https://polygonscan.com/address/0xC52329f4164E115ca40B3D5Ab8E1FF3F85E9A253 **(FAANGOrchestrator)**
Timelock	https://polygonscan.com/address/0x362e7d903359e5649965e12b90f3d5abbee698b9

1.5 Issues on Polygon

	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High Severity	0	0	0	0
Medium Severity	0	0	0	0
Low Severity	6	0	0	6
Total	**6**	0	0	6

Detail Issues on Polygon:
#1

Severity: **Low**

Contract: **FAANGToken**

Function name: **getPriorVotes**(address,uint256)

Estimated Gas Usage: **309 – 404**

Log:

In file: FAANGOrchestrator.sol:1179

require(blockNumber < block.number, "FAANG::getPriorVotes: not yet determined")

Initial State:

Account: [CREATOR], balance: 0x40000000000000, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [SOMEGUY], function: getPriorVotes(address,uint256), txdata: 0x782d6fe1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000, value: 0x0

A control flow decision is made based on the block.number environment variable.

The block.number environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

#2

Severity: **Low**

Contract: **FAANGToken**

Function name: **delegateBySig**(address,uint256,uint256,uint8,bytes32,bytes32)

Estimated Gas Usage: **13865 – 72108**

Log:

In file: FAANGOrchestrator.sol:1149

require(block.timestamp <= expiry, "FAANG::delegateBySig: signature expired")

Initial State:

Account: [CREATOR], balance: 0x2000000, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [ATTACKER], function: delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32), txdata:
0xc3cda5200000000000000000000000000000202020000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000, value: 0x0

A control flow decision is made based on the block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

#3

Severity: Low

Contract: **FAANGToken**

Function name: **delegate**(address)

Estimated Gas Usage: **11500 – 36374**

Log:

In file: FAANGOrchestrator.sol:1265

*require(n < 2**32, errorMessage)*

Initial State:

Account: [CREATOR], balance: 0x1, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [CREATOR], function: mint(address,uint256), txdata: 0x40c10f19000000000000000000000000affeaffeaffeaffeaffeaffeaffeaffeaffeaffe00000000000000000000000000000000000000000000000001000000000000000, value: 0x0

Caller: [CREATOR], function: delegate(address), txdata: 0x5c19a95c101010101010101010101010000000000000000000000000000000000000000001, value: 0x0

A control flow decision is made based on the block.number environment variable.

The block.number environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

#4

Severity: Low

Contract: **FAANGToken**

Function name: **getPriorVotes**(address,uint256)

Estimated Gas Usage: **309 – 404**

Log:

In file: TEXOOrchestrator.sol:1179

require(blockNumber < block.number, "TEXO::getPriorVotes: not yet determined")

Initial State:

Account: [CREATOR], balance: 0x0, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [CREATOR], function: getPriorVotes(address,uint256), txdata: 0x782d6fe1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000, value: 0x0

A control flow decision is made based on the block.number environment variable.

The block.number environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

#5

Severity: Low

Contract: **TEXOToken**

Function name: **delegateBySig**(address,uint256,uint256,uint8,bytes32,bytes32)

Estimated Gas Usage: **13865 – 72108**

Log:

In file: TEXOOrchestrator.sol:1149

require(block.timestamp <= expiry, "TEXO::delegateBySig: signature expired")

Initial State:

Account: [CREATOR], balance: 0x2000000, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [SOMEGUY], function: delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32), txdata:
0xc3cda52000000000000000000000000000000000000000000000000100000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000, value: 0x0

A control flow decision is made based on the block.timestamp environment variable.

The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

#6

Severity: **Low**

Contract: **TEXOToken**

Function name: **delegate**(address)

Estimated Gas Usage: **11500 – 36374**

Log:

In file: TEXOOrchestrator.sol:1265

*require(n < 2**32, errorMessage)*

Initial State:

Account: [CREATOR], balance: 0x0, nonce:0, storage:{}

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}

Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], calldata: , value: 0x0

Caller: [CREATOR], function: mint(address,uint256), txdata: 0x40c10f190000000000000000000000affeaffeaffeaffeaffeaffeaffeaffeaffe0000c000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000, value: 0x0

Caller: [CREATOR], function: delegate(address), txdata: 0x5c19a95c0000000000000000000000000000000000000000000000000000000000000001, value: 0x0

A control flow decision is made based on the block.number environment variable.

The block.number environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks.

Resolution: Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

2 Technical Analyzing

2.1 TEXOToken

The TEXOToken is a simple token with basic functionality. It allows for tEXO tokens to be minted when the **mint** function is called by the Owner prior to ownership being transferred to the Masterchef. This can be used to pre-mint tokens for initial liquidity. After ownership is transferred to the Masterchef, only the Masterchef will be able to mint tokens.

At the time of this report, 100 tEXO was pre-minted to the address 0x36dc08Fe2286D39021FeD05A6CCB94a2A7a86fA2 (an EOA).

2.1.1 Token Overview

Address	0xF1afb5674Bf946458BD1163163F62dE683B07D65
Token Supply	Unlimited
Decimal Places	18
Max Transfer	Unlimited
Min Transfer	0
Transfer Fee	0

2.1.2 Privileged Roles

The owner of the token contract has been transfer to the Masterchef. The owner is able to call the following owner-only functions:

- Mint

2.2 tEXO Masterchef

The Masterchef has a maximum deposit fee of 4%, and a maximum referral commission rate of 20%. The referral rate is initially set at 2%, and can be changed by the owner with the function **updateReferralBonusBp**. The initial emission rate is 0.5 tEXO per block, which can be reduced to no lower 0.1 tEXO per block on BSC and 0.35 tEXO per block, with the minium of 0.07 tEXO per block on Polygon. The emission would be reduced for every 28800 blocks on BSC with the reduction rate of 15%; on Polygon the emission is reduced every 43200 blocks with the rate of 15%. The emission rate can be manually updated by the owner with the function **setEmissionRate**. Finally, the **blockToStartReducingEmissionRate** can be changed with the function **setBlockToStartReducingEmissionRate** by the owner.

2.2.1 Privileged Roles

The following function can only be called by the owner of the address:

- add
- set
- setEmissionRate
- setBlockToStartReducingEmissionRate
- updateReferralBonusBp

2.3 FAANGToken

The FAANGToken is a yield token, with the basic functionality of a standard token. The token can be minted through the function **mint** by the owner address. No token was pre-minted before the ownership transfer to the FAANG Masterchef

2.3.1 Token Overview

Address	0x2b0EA770431df39196FF33fB425BE084636d946e
Token Supply	Unlimited
Decimal Places	18
Max Transfer	Unlimited
Min Transfer	0
Transfer Fee	0

2.3.2 Privileged Roles

The following functions can be called only by the owner of the token contract:

- mint

2.4 FAANG Masterchef

The Masterchef has a fixed emission rate of 1 FAANG token per block on BSC and 0.7 FAANG token per block on Polygon. Also, the **startBlock** can be updated via the function **updateStartBlock** by the contract owner

2.4.1 Privileged Roles

The following functions can be called only by the owner of the token contract:

- add
- set
- updateStartBlock

2.5 Timelock

The Timelock is a standard timelock contract with no additional features.

Website

Email: hello@bcfg.io