Cryptonics

Audit Report

Gath3r Smart Contracts

September 8, 2020

Table of Contents

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

This audit has been performed by

Stefan Beyer

Cryptonics Consulting S.L.
Ramiro de Maeztu 7
46022 Valencia
SPAIN

https://cryptonics.consulting/
info@cryptonics.consulting

Summary of Findings

No issues have been found in the latest version of the smart contracts submitted for this audit.

Introduction

Purpose of this Report

Cryptonics Consulting has been engaged to perform a smart contract audit for Gath3r (https://gather.network/)..

The objectives of the audit are as follows:

1. Determine correct functioning of the contract, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine contract bugs, which might lead to unexpected behavior.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents the summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The code has been provided by the developers in the form of zip archive file with the following SHA256 hash value:

```
86b49b3e37f7841efbf6a8cf7f8d9d771d2b20f4c878ea72be92b09d17f9625b
```

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the contract's intended purpose by reading the available documentation.
2. Automated scanning of the contract with static code analysis tools for security vulnerabilities and use of best practice guidelines.
3. Manual line by line analysis of the contracts source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    - Reentrancy analysis
    - Race condition analysis
    - Front-running issues and transaction order dependencies
    - Time dependencies
    - Under- / overflow and math precision issues
    - Function visibility Issues
    - Possible denial of service attacks
    - Storage Layout Vulnerabilities
4. Report preparation

Project Overview

Intended Functionality

The submitted code implements a mintable ERC20 token, a vestin pool and a multisig ownership scheme.

Smart Contracts Audited

The following files have been covered in the audit process:

```
contracts
├─── Gather_coin.sol
├─── Migrations.sol
├─── Ownable.sol
├─── SafeMath.sol
├─── TokenController.sol
├─── VestingPool.sol
├─── VestingPoolController.sol
└─── multiowned.sol
```

Detailed Findings

Issues Encountered

No issues were encountered in the latest version of the smart contracts submitted for this audit.

Overall Code Quality

The overall quality of the code submitted for the audit is excellent.

Best practise recommendations have largely been followed. Existing, audited code has been used whenever possible in the form of the Openzeppellin libraries (https://openzeppelin.com/).

A safe math library has been used for arithmetic operations to avoid overflow and underflow issues.

Code layout mostly follows the official Solidity style guide (https://solidity.readthedocs.io/en/v0.6.11/style-guide.html).

Appendix - Functions

```
+ Contract ERC20Basic
  - From ERC20Basic
    - balanceOf(address) (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)

+ Contract ERC20
  - From ERC20Basic
    - balanceOf(address) (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)
  - From ERC20
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)

+ Contract BasicToken
  - From BasicToken
    - balanceOf(address) (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)

+ Contract StandardToken
  - From BasicToken
    - balanceOf(address) (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)
  - From StandardToken
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - decreaseApproval(address,uint256) (public)
    - increaseApproval(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)

+ Contract MintableToken
  - From Ownable
    - constructor() (public)
    - transferOwnership(address) (public)
  - From BasicToken
    - balanceOf(address) (public)
    - totalSupply() (public)
    - transfer(address,uint256) (public)
  - From StandardToken
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - decreaseApproval(address,uint256) (public)
    - increaseApproval(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)
  - From MintableToken
    - checkMintPermission(address) (private)
```

```
    - finishMinting() (public)
    - mint(address,uint256) (public)
    - mintAllowed(address) (public)
    - mintInternal(address,uint256) (internal)
    - setMinter(address,uint256) (public)

+ Contract GatherToken (Most derived contract)
  - From Ownable
    - transferOwnership(address) (public)
  - From BasicToken
    - balanceOf(address) (public)
    - totalSupply() (public)
  - From StandardToken
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - decreaseApproval(address,uint256) (public)
    - increaseApproval(address,uint256) (public)
  - From MintableToken
    - checkMintPermission(address) (private)
    - finishMinting() (public)
    - mint(address,uint256) (public)
    - mintAllowed(address) (public)
    - mintInternal(address,uint256) (internal)
    - setMinter(address,uint256) (public)
  - From GatherToken
    - constructor() (public)
    - pauseTransfer() (public)
    - resumeMinting(uint256) (public)
    - sendTokens(address[],uint256[]) (public)
    - transfer(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)
    - unpauseTransfer() (public)

+ Contract Ownable
  - From Ownable
    - constructor() (public)
    - transferOwnership(address) (public)

+ Contract SafeMath (Most derived contract)
  - From SafeMath
    - add(uint256,uint256) (internal)
    - div(uint256,uint256) (internal)
    - mul(uint256,uint256) (internal)
    - sub(uint256,uint256) (internal)

+ Contract VestingPool (Most derived contract)
  - From Ownable
    - constructor() (public)
    - transferOwnership(address) (public)
  - From VestingPool
    - _amountWithPrecision(uint256) (internal)
    - _expandToDecimals(uint256) (internal)
    - _expandToDecimalsVestingScheme(bytes32) (internal)
    - _initVestingData() (internal)
    - _resetAllAdminApprovals(address) (internal)
    - _withdraw(address,uint256,bytes32) (internal)
    - addAdmin1address(address) (public)
```

```
    - addAdmin2address(address) (public)
    - constructor(address) (public)
    - emergencyTransferFor(bytes32,address) (public)
    - firstAdminEmergencyApproveFor(bytes32,address) (public)
    - firstMultiownedEmergencyApproveFor(bytes32) (public)
    - getAvailableAmountFor(bytes32) (public)
    - multipleWithdraw(address[],uint256[],bytes32) (public)
    - secondAdminEmergencyApproveFor(bytes32,address) (public)
    - secondMultiownedEmergencyApproveFor(bytes32) (public)
    - startVesting() (public)



+ Contract GatherToken (Most derived contract)
  - From Ownable
    - transferOwnership(address) (public)
  - From BasicToken
    - balanceOf(address) (public)
    - totalSupply() (public)
  - From StandardToken
    - allowance(address,address) (public)
    - approve(address,uint256) (public)
    - decreaseApproval(address,uint256) (public)
    - increaseApproval(address,uint256) (public)
  - From MintableToken
    - checkMintPermission(address) (private)
    - finishMinting() (public)
    - mint(address,uint256) (public)
    - mintAllowed(address) (public)
    - mintInternal(address,uint256) (internal)
    - setMinter(address,uint256) (public)
  - From GatherToken
    - constructor() (public)
    - pauseTransfer() (public)
    - resumeMinting(uint256) (public)
    - sendTokens(address[],uint256[]) (public)
    - transfer(address,uint256) (public)
    - transferFrom(address,address,uint256) (public)
    - unpauseTransfer() (public)


+ Contract TokenController (Most derived contract)
  - From multiowned
    - addOwner(address) (external)
    - amIOwner() (external)
    - assertOperationIsConsistent(bytes32) (private)
    - assertOperationIsConsistentForAll(bytes32) (private)
    - assertOwnersAreConsistent() (private)
    - changeOwner(address,address) (external)
    - changeRequirement(uint256) (external)
    - checkOwnerIndex(uint256) (private)
    - clearPending() (private)
    - confirmAndCheck(bytes32) (private)
    - confirmAndCheckForAll(bytes32) (private)
    - constructor(address[],uint256) (public)
    - getOwner(uint256) (public)
```

```
    - getOwners() (public)
    - hasConfirmed(bytes32,address) (external)
    - isOperationActive(bytes32) (private)
    - isOwner(address) (public)
    - makeOwnerBitmapBit(address) (private)
    - removeOwner(address) (external)
    - reorganizeOwners() (private)
    - revoke(bytes32) (external)
  - From TokenController
    - constructor(address[],uint256,address) (public)
    - finishMinting() (public)
    - mint(address,uint256) (public)
    - pauseTransfer() (public)
    - resumeMinting(uint256) (public)
    - sendTokens(address[],uint256[]) (public)
    - setMinter(address,uint256) (public)
    - transfer(address,uint256) (public)
    - transferOwnership(address) (public)
    - unpauseTransfer() (public)

+ Contract multiowned
  - From multiowned
    - addOwner(address) (external)
    - amIOwner() (external)
    - assertOperationIsConsistent(bytes32) (private)
    - assertOperationIsConsistentForAll(bytes32) (private)
    - assertOwnersAreConsistent() (private)
    - changeOwner(address,address) (external)
    - changeRequirement(uint256) (external)
    - checkOwnerIndex(uint256) (private)
    - clearPending() (private)
    - confirmAndCheck(bytes32) (private)
    - confirmAndCheckForAll(bytes32) (private)
    - constructor(address[],uint256) (public)
    - getOwner(uint256) (public)
    - getOwners() (public)
    - hasConfirmed(bytes32,address) (external)
    - isOperationActive(bytes32) (private)
    - isOwner(address) (public)
    - makeOwnerBitmapBit(address) (private)
    - removeOwner(address) (external)
    - reorganizeOwners() (private)
    - revoke(bytes32) (external)


+ Contract VestingPoolController (Most derived contract)
  - From multiowned
    - addOwner(address) (external)
    - amIOwner() (external)
    - assertOperationIsConsistent(bytes32) (private)
    - assertOperationIsConsistentForAll(bytes32) (private)
    - assertOwnersAreConsistent() (private)
    - changeOwner(address,address) (external)
    - changeRequirement(uint256) (external)
    - checkOwnerIndex(uint256) (private)
    - clearPending() (private)
    - confirmAndCheck(bytes32) (private)
```

```
  - confirmAndCheckForAll(bytes32) (private)
  - constructor(address[],uint256) (public)
  - getOwner(uint256) (public)
  - getOwners() (public)
  - hasConfirmed(bytes32,address) (external)
  - isOperationActive(bytes32) (private)
  - isOwner(address) (public)
  - makeOwnerBitmapBit(address) (private)
  - removeOwner(address) (external)
  - reorganizeOwners() (private)
  - revoke(bytes32) (external)
- From VestingPoolController
  - addAdmin1(address) (public)
  - addAdmin2(address) (public)
  - advisorWithdraw(address[],uint256[]) (public)
  - constructor(address[],uint256,address) (public)
  - emergencyPublicTransferFor(bytes32,address) (public)
  - firstAdminEmergencyApproveFor(bytes32) (public)
  - firstMultiownedEmergencyApproveFor(bytes32) (public)
  - foundationWithdraw(address[],uint256[]) (public)
  - getAvailableAmountForAdvisor() (public)
  - getAvailableAmountForFoundation() (public)
  - getAvailableAmountForMarketing() (public)
  - getAvailableAmountForPlatform() (public)
  - getAvailableAmountForPrivate() (public)
  - getAvailableAmountForPublic() (public)
  - getAvailableAmountForSeed() (public)
  - getAvailableAmountForTeam() (public)
  - marketingWithdraw(address[],uint256[]) (public)
  - platformWithdraw(address[],uint256[]) (public)
  - privateWithdraw(address[],uint256[]) (public)
  - publicWithdraw(address[],uint256[]) (public)
  - secondAdminEmergencyApproveFor(bytes32) (public)
  - secondMultiownedEmergencyApproveFor(bytes32) (public)
  - seedWithdraw(address[],uint256[]) (public)
  - startVesting() (public)
  - teamWithdraw(address[],uint256[]) (public)
  - transferOwnership(address) (public)
```