Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

Gilgamesh

Deployer address

0x4cafec4e6dbd1fe134ba45d288fe9cc01ab71352

Client contacts:

Gilgamesh team

Blockchain

Ethereum

Project website:

https://www.gilgamesheth.org

Disclaimer

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

DISCLAIMER: 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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate 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 TechRate 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, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, 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 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.

Background

TechRate was commissioned by Gilgamesh to perform an audit of smart contracts:
https://etherscan.io/address/0xfde19f0de7a4e7eca8ab29c9f202a21a3b3503de#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

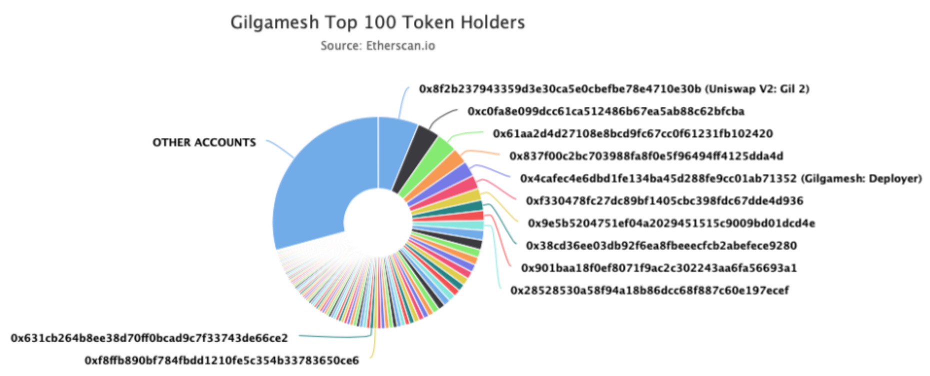
Token contract details for 10.11.2021

Contract name	Gilgamesh
Contract address	0xfdE19f0de7a4E7ECa8AB29c9f202A21A3B3503De
Total supply	1,000,000,000,000,000,000
Token ticker	Gil
Decimals	9
Token holders	1,851
Transactions count	4,162
Top 100 holders dominance	70.78%
Contract deployer address	0x4cafec4e6dbd1fe134ba45d288fe9cc01ab71352
Contract's current owner address	0x0000000000000000000000000000000000000000

Gilgamesh Token Distribution

The top 100 holders collectively own 70.78% (707,819,207,546,901,000.00 Tokens) of Gilgamesh

Token Total Supply: 1,000,000,000,000,000,000.00 Token | Total Token Holders: 1,851

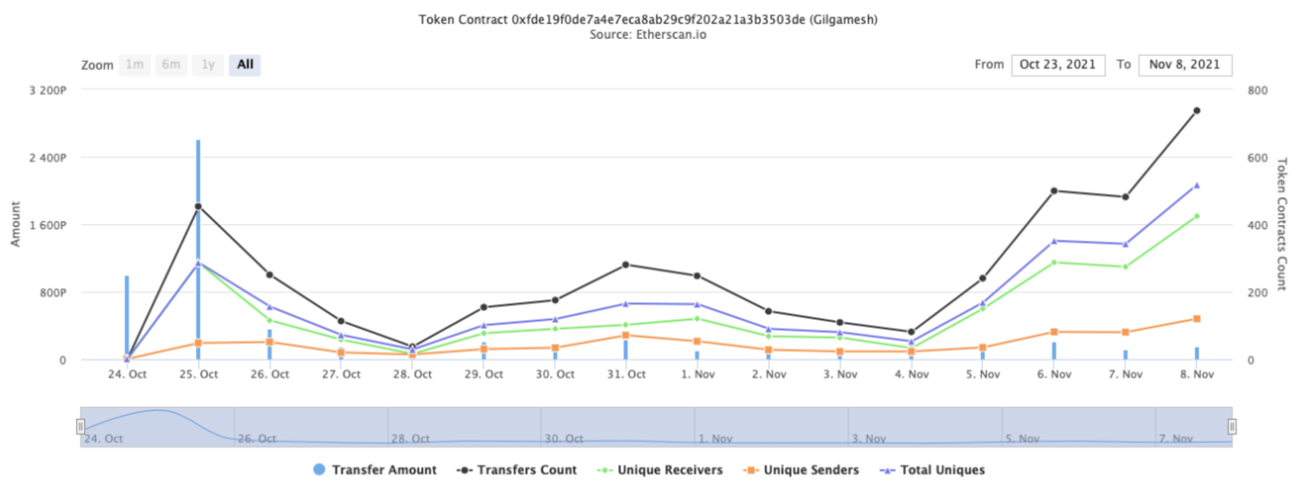


(A total of 707,819,207,546,901,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000,000.00 token)

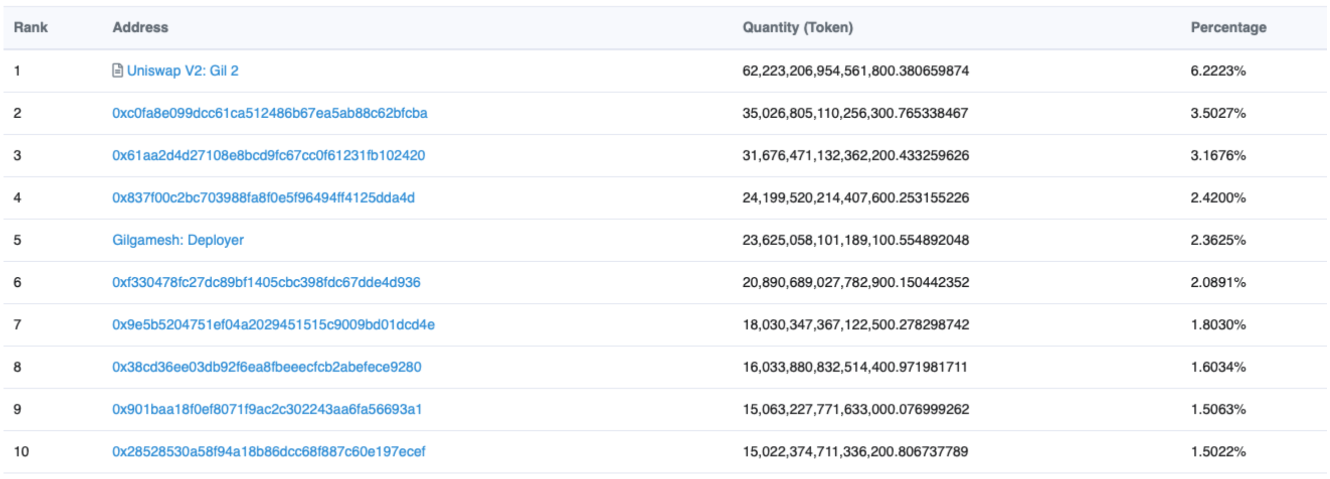
Gilgamesh Contract interaction details

Time Series: Token Contract Overview

Sun 24, Oct 2021 - Mon 8, Nov 2021



Gilgamesh Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	Uniswap V2: Gil 2	62,223,206,954,561,800.380659874	6.2223%
2	0xc0fa8e099dcc61ca512486b67ea5ab88c62bfcba	35,026,805,110,256,300.765338467	3.5027%
3	0x61aa2d4d27108e8bcd9fc67cc0f61231fb102420	31,676,471,132,362,200.433259626	3.1676%
4	0x837f00c2bc703988fa8f0e5f96494ff4125dda4d	24,199,520,214,407,600.253155226	2.4200%
5	Gilgamesh: Deployer	23,625,058,101,189,100.554892048	2.3625%
6	0xf330478fc27dc89bf1405cbc398fdc67dde4d936	20,890,689,027,782,900.150442352	2.0891%
7	0x9e5b5204751ef04a2029451515c9009bd01dcd4e	18,030,347,367,122,500.278298742	1.8030%
8	0x38cd36ee03db92f6ea8fbeeecfcb2abefece9280	16,033,880,832,514,400.971981711	1.6034%
9	0x901baa18f0ef8071f9ac2c302243aa6fa56693a1	15,063,227,771,633,000.076999262	1.5063%
10	0x28528530a58f94a18b86dcc68f887c60e197ecef	15,022,374,711,336,200.806737789	1.5022%

Contract functions details

+ Context
  - [Int] _msgSender

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner

+ [Int] IUniswapV2Factory
  - [Ext] createPair #

+ [Int] IUniswapV2Router02
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidityETH ($)

+ Gilgamesh (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Ext] setCooldownEnabled #
    - modifiers: onlyOwner
  - [Prv] tokenFromReflection
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] swapTokensForEth #

    - modifiers: lockTheSwap
  - [Prv] sendETHToFee #
  - [Ext] openTrading #
    - modifiers: onlyOwner
  - [Pub] setBots #
    - modifiers: onlyOwner
  - [Pub] delBot #
    - modifiers: onlyOwner
  - [Prv] _tokenTransfer #
  - [Prv] _transferStandard #
  - [Prv] _takeTeam #
  - [Prv] _reflectFee #
  - [Ext] <Fallback> ($)
  - [Ext] manualswap #
  - [Ext] manualsend #
  - [Prv] _getValues
  - [Prv] _getTValues
  - [Prv] _getRValues
  - [Prv] _getRate
  - [Prv] _getCurrentSupply
  - [Ext] migrateHolders #
    - modifiers: onlyOwner

($) = payable function
# = non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

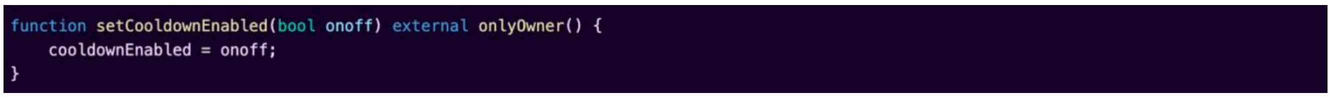
No medium severity issues found.

Low Severity Issues

No low severity issues found.

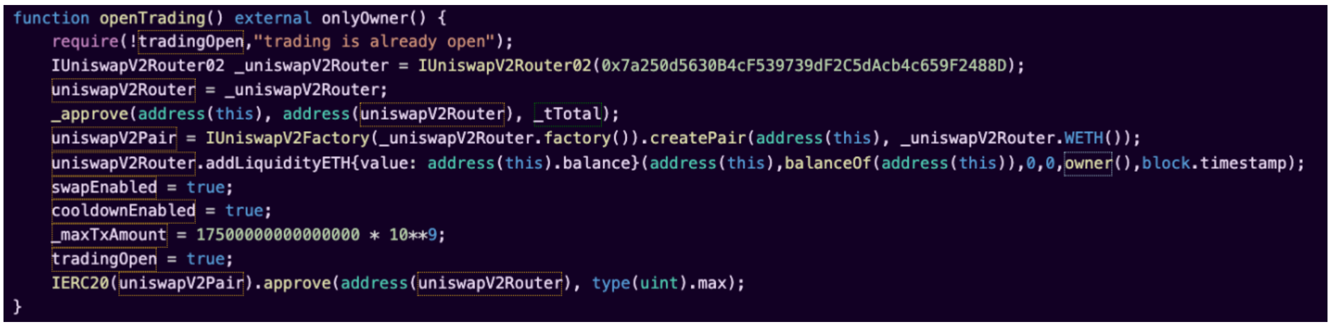
Owner privileges (In the period when the owner is not renounced)

- Owner can enable cooldown (user to user trading with time offset).



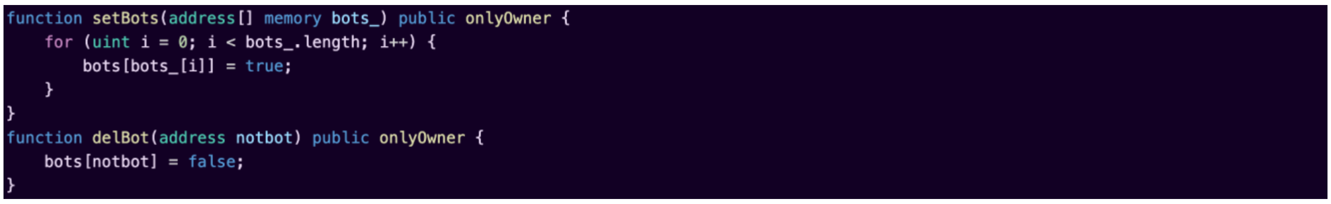
```
function setCooldownEnabled(bool onoff) external onlyOwner() {
    cooldownEnabled = onoff;
}
```

- Owner can open swap trading.



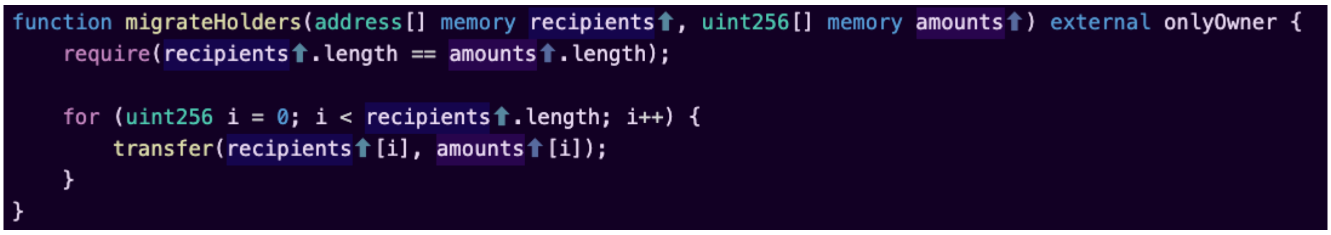
```
function openTrading() external onlyOwner() {
    require(!tradingOpen,"trading is already open");
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D);
    uniswapV2Router = _uniswapV2Router;
    _approve(address(this), address(uniswapV2Router), _tTotal);
    uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this), _uniswapV2Router.WETH());
    uniswapV2Router.addLiquidityETH{value: address(this).balance}(address(this),balanceOf(address(this)),0,0,owner(),block.timestamp);
    swapEnabled = true;
    cooldownEnabled = true;
    _maxTxAmount = 17500000000000000 * 10**9;
    tradingOpen = true;
    IERC20(uniswapV2Pair).approve(address(uniswapV2Router), type(uint).max);
}
```

- Owner can add and remove bots (no transferring between this addresses).



```
function setBots(address[] memory bots_) public onlyOwner {
    for (uint i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}
function delBot(address notbot) public onlyOwner {
    bots[notbot] = false;
}
```

- **Owner can multiple transfer.**



```
function migrateHolders(address[] memory recipients, uint256[] memory amounts) external onlyOwner {
    require(recipients.length == amounts.length);

    for (uint256 i = 0; i < recipients.length; i++) {
        transfer(recipients[i], amounts[i]);
    }
}
```

Conclusion

Smart contracts do not contain high severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details are NOT provided by the team.

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.