HackSafe

Smart Contract Security Audit Report

Fatty Foodles

May 2022

Security Status

★★★★½

www.hacksafe.io

Audit Details

Audited project
Fatty Foodles

Deployer address
0x9B8F11A2784c62F3363142029e5dB62BDc4f7fC1

Client contacts
Fatty Foodles team

Blockchain
Ethereum

Website
Not provided by team

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) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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

HeckSafe was commissioned by Fatty Foodles to perform an audit of smart contracts:

- https://etherscan.io/address/0x05C89e762A802652a61521fA8e42593cC498c007#code

Contract Details

Token contract details for 07.05.2022

Contract name	: FattyFoodles
Contract address	: 0x05C89e762A802652a61521fA8e42593cC498c007
Token-Type	: ERC721
Total supply	: 2, 000
Token Ticker	: FOOD
Token Holders	: 422
Transactions count	: 2,383
Contract deployer address	: 0x9B8F11A2784c62F3363142029e5dB62BDc4f7fC1

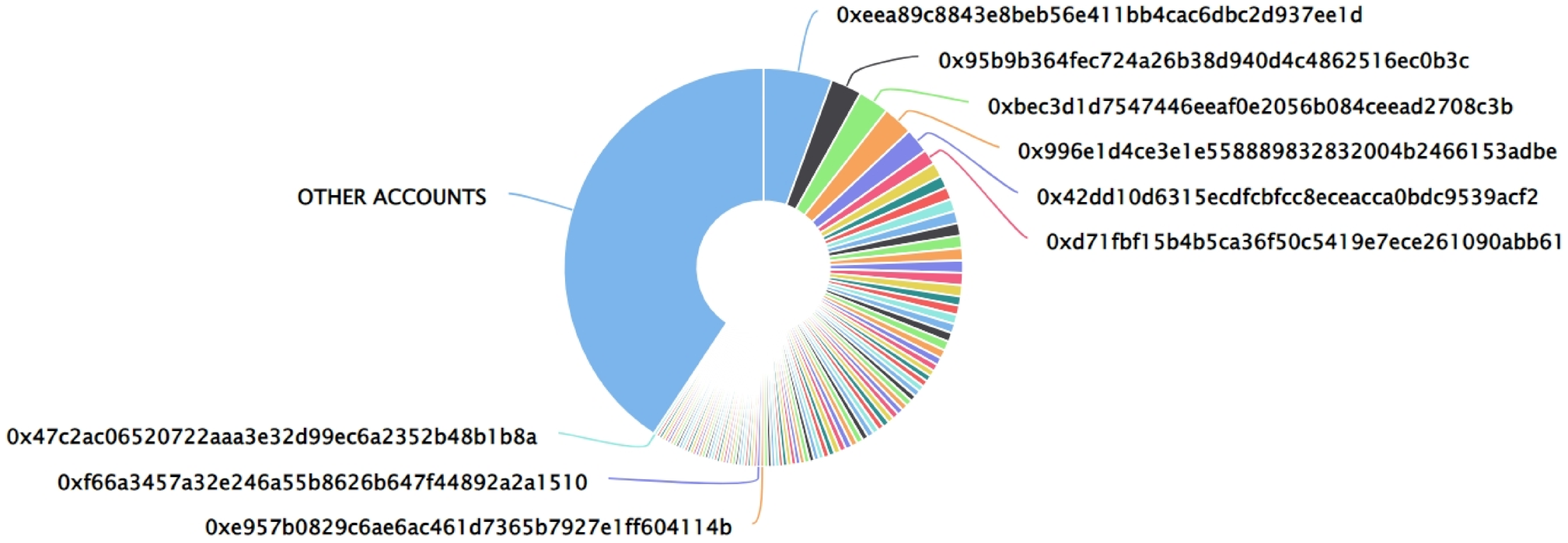
Fatty Foodles Token Distribution

The top 100 holders collectively own 59.30% (1,186.00 Tokens) of Fatty Foodles

Token Total Supply: 2,000.00 Token | Total Token Holders: 422

Fatty Foodles Top 100 Token Holders

Source: Etherscan.io



Fatty Foodles Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0xeea89c8843e8beb56e411bb4cac6dbc2d937ee1d	112	5.6000%
2	*0x🐺.eth	50	2.5000%
3	stilpo.eth	50	2.5000%
4	*the-barn.eth	49	2.4500%
5	omun.eth	40	2.0000%
6	0xd71fbf15b4b5ca36f50c5419e7ece261090abb61	25	1.2500%
7	dad.eth	23	1.1500%
8	0x00f09f87a2e15c9666f37ad48c01e180e66381ed	20	1.0000%
9	qualityblockshop.eth	20	1.0000%
10	0x09da17ab73f904a755fad7160d529be57a39d3dd	20	1.0000%

Contract functions details

+ [Lib] Strings
    -[Int] toString #
    -[Int] toHexString #
    -[Int] toHexString #

+ Context
    -[Int] _msgSender
    -[Int] _msgData

+ Ownable (Context)
    -< constructor>
    -[Pub] owner
    -[Pub] renounceOwnership#
        -Modifier: onlyOwner
    -[Pub] transferOwnership#
        -Modifier: onlyOwner
    -[Pvt] _setOwner #

-[Lib] Address
    -[Int] isContract
    -[Int] sendValue
    -[Int] functionCall
    -[Int] functionCall
    -[Int] functionCallWithValue
    -[Int] functionCallWithValue
    -[Int] functionStaticCall
    -[Int] functionStaticCall
    -[Int] functionDelegateCall
    -[Int] functionDelegateCall
    -[Int] verifyCallResult

+ [Int] IERC721Receiver
    -[Ext] onERC721Received

+ [Int] IERC165
    -[Ext] supportsInterface

+ ERC165 (IERC165)
    -[Pub] supportsInterface

Contract functions details

+ [Int] IERC721 (IERC165)
    -[Ext] balanceOf
    -[Ext] ownerOf
    -[Ext] safeTransferFrom #
    -[Ext] transferFrom #
    -[Ext] approve #
    -[Ext] getApproved
    -[Ext] setApprovalForAll
    -[Ext] isApprovedForAll
    -[Ext] safeTransferFrom

+ [Int] IERC721Enumerable (IERC721)
    -[Ext] totalSupply
    -[Ext] tokenOfOwnerByIndex
    -[Ext] tokenByIndex

+ [Int] IERC721Metadata (IERC721)
    -[Ext] name
    -[Ext] symbol
    -[Ext] tokenURI

+ ERC721 (Context, ERC165, IERC721, IERC721Metadata)
    -< constructor >
    -[Pub] supportsInterface
    -[Pub] balanceOf
    -[Pub] ownerOf
    -[Pub] name
    -[Pub] symbol
    -[Pub] tokenURI #
    -[Int] _baseURI
    -[Pub] approve #
    -[Pub] getApproved #
    -[Pub] setApprovalForAll #
    -[Pub] isApprovedForAll #
    -[Pub] transferFrom #
    -[Pub] safeTransferFrom #
    -[Pub] safeTransferFrom #
    -[Int] _safeTransfer #
    -[Int] _exists

Contract functions details

-[Int] _isApprovedOrOwner
-[Int] _safeMint #
-[Int] _safeMint #
-[Int] _mint #
-[Int] _burn #
-[Int] _transfer #
-[Int] _approve #
-[Pvt] _checkOnERC721Received
-[Int] _beforeTokenTransfer

+ ERC721Enumerable (ERC721, IERC721Enumerable)
-[Pub] supportsInterface
-[Pub] tokenOfOwnerByIndex
-[Pub] totalSupply
-[Pub] tokenByIndex
-[Int] _beforeTokenTransfer #
-[Pvt] _addTokenToOwnerEnumeration #
-[Pvt] _addTokenToAllTokensEnumeration #
-[Pvt] _removeTokenFromOwnerEnumeration #
-[Pvt] _removeTokenFromAllTokensEnumeration #

+ FattyFoodles (ERC721Enumerable, Ownable)
-<constructor>
-[Int] _baseURI
-[Pub] mint $
-[Pub] walletOfOwner #
-[Pub] tokenURI
-[Pub] setCost #
-modifiers: onlyOwner
-[Pub] setBaseURI #
-modifiers: onlyOwner
-[Pub] setBaseExtension #
-modifiers: onlyOwner
-[Pub] pause #
-modifiers: onlyOwner
-[Pub] whitelistUser #
-modifiers: onlyOwner

Contract functions details

-[Pub] removeWhitelistUser #
  -modifiers: onlyOwner
-[Pub] withdraw#
  -modifiers: onlyOwner

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	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.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Low issue
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

- **Critical Severity Issues**
  No critical severity issue found.

- **High Severity Issues**
  No high severity issue found.

- **Medium Severity Issues**
  No medium severity issues found.

- **Low Severity Issues**
  Two low severity issues found.

1. Unlocked Compiler Version.

- **Description**

  The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

  We advise to use only one compiler version instead multi pragma which is alternatively locked at the lowest version possible so that the contract can be compiled. Use following line instead of pragma solidity ^0.8.0;

  pragma solidity 0.8.0;

2. Scoping and Declarations.

Unused function.

- **Description**

  The functions declared line number 64, 80, 114, 247, 272, 282, 301, 315, 334, 344, 361, 371, 388 do nothing.

- **Location:**

  Line no: 64, 80, 114, 247, 272, 282, 301, 315, 334, 344, 361, 371, 388.

- **Recommendation:**

  We advise to remove unused code which can help you to develop clean coding style and save some computational gas too.

Owner Privileges

Owner Privileges (in the period when the owner is not renounced) :

- Fatty Foodles Contract:
    - Owner can transfer ownership.
    - Owner can renounce ownership.
    - Owner can set cost of minting nft.
    - Owner can set maximum amount.
    - Owner can update base URI.
    - Owner can update base extension.
    - Owner can pause transfers.
    - Owner can add minter.
    - Owner can remove minter.
    - Owner can withdraw funds.

Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe 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.