HackSafe

Smart Contract Security Audit Report

Sushi

September 2022

Security Status

★★★⯪☆

www.hacksafe.io

Audit Details

Audited project
Sushi

Deployer address
0x2D407dDb06311396fE14D4b49da5F0471447d45C

Client contacts
sushi Team

Blockchain
Fantom

Website
https://sushiswap.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) (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.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by sushi to perform an audit of smart contracts:

- https://ftmscan.com/address/0xae75A438b2E0cB8Bb01Ec1E1e376De11D44477CC#code

The purpose of the audit was to achieve the

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

The information in this report should be 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.

Contract Details

Token contract details for 23.09.2022

Token Type	: ERC20
Contract name	: Sushi
Contract address	: 0xae75A438b2E0cB8Bb01Ec1E1e376De11D44477CC
Total supply	: 622,663.936148
Token ticker	: SUSHI
Decimals	: 18
Token holders	: 2,191
Transactions count	: 967,154
Compiler version	: v0.8.1+commit.df193b15
Contract deployer address	: 0x2D407dDb06311396fE14D4b49da5F0471447d45C
Owner address	: 0xc564ee9f21ed8a2d8e7e76c085740d5e4c5fafbe

Social profiles

Twitter Profile	: https://twitter.com/sushiswap
Coinmarketcap Profile	: https://coinmarketcap.com/currencies/sushiswap/
Coingecko profile	: https://www.coingecko.com/en/coins/sushi/

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are "poor secure". This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here

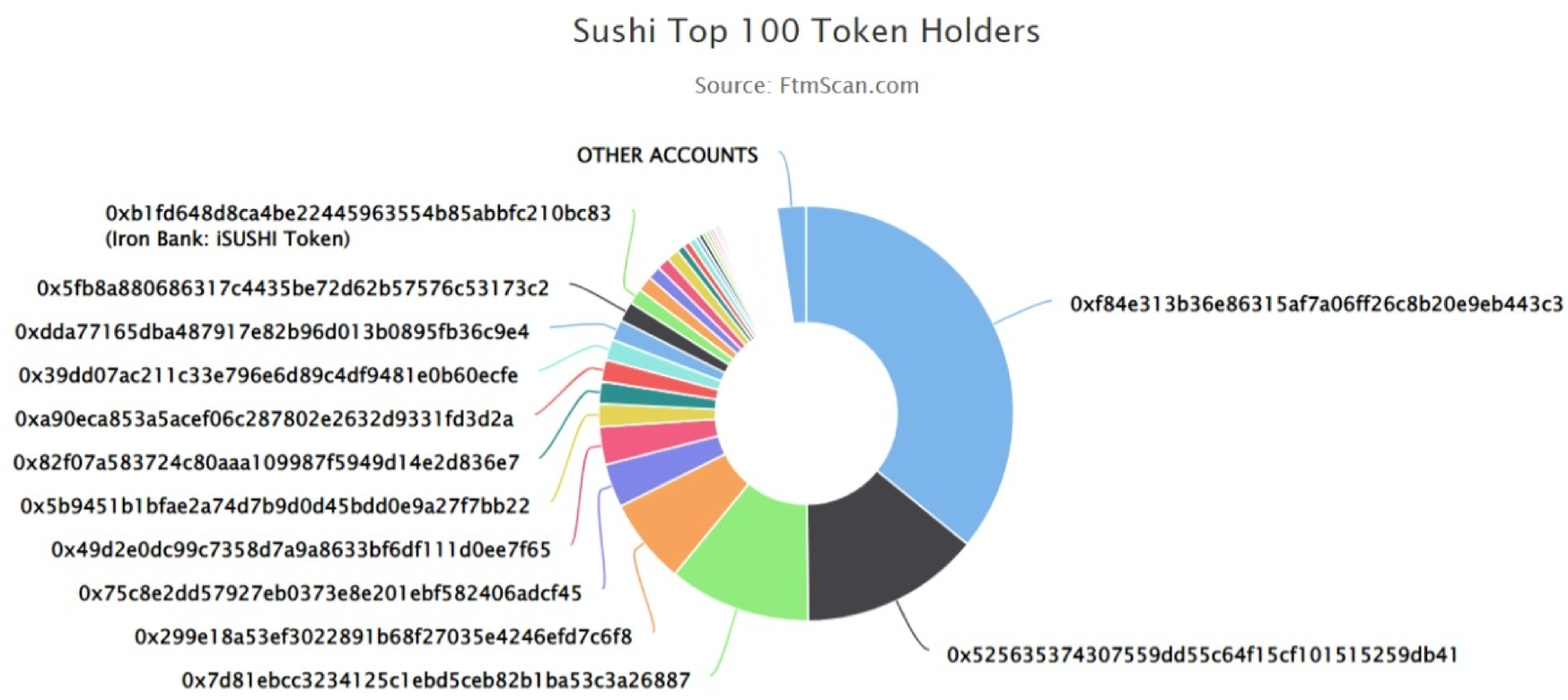
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 1 high, 0 medium and 0 low and some very low-level issues. These issues are not critical ones.

Sushi Token Distribution

The top 100 holders collectively own 97.74% (608,561.19 Tokens) of Sushi

Token Total Supply: 622,663.94 Token | Total Token Holders: 2,191



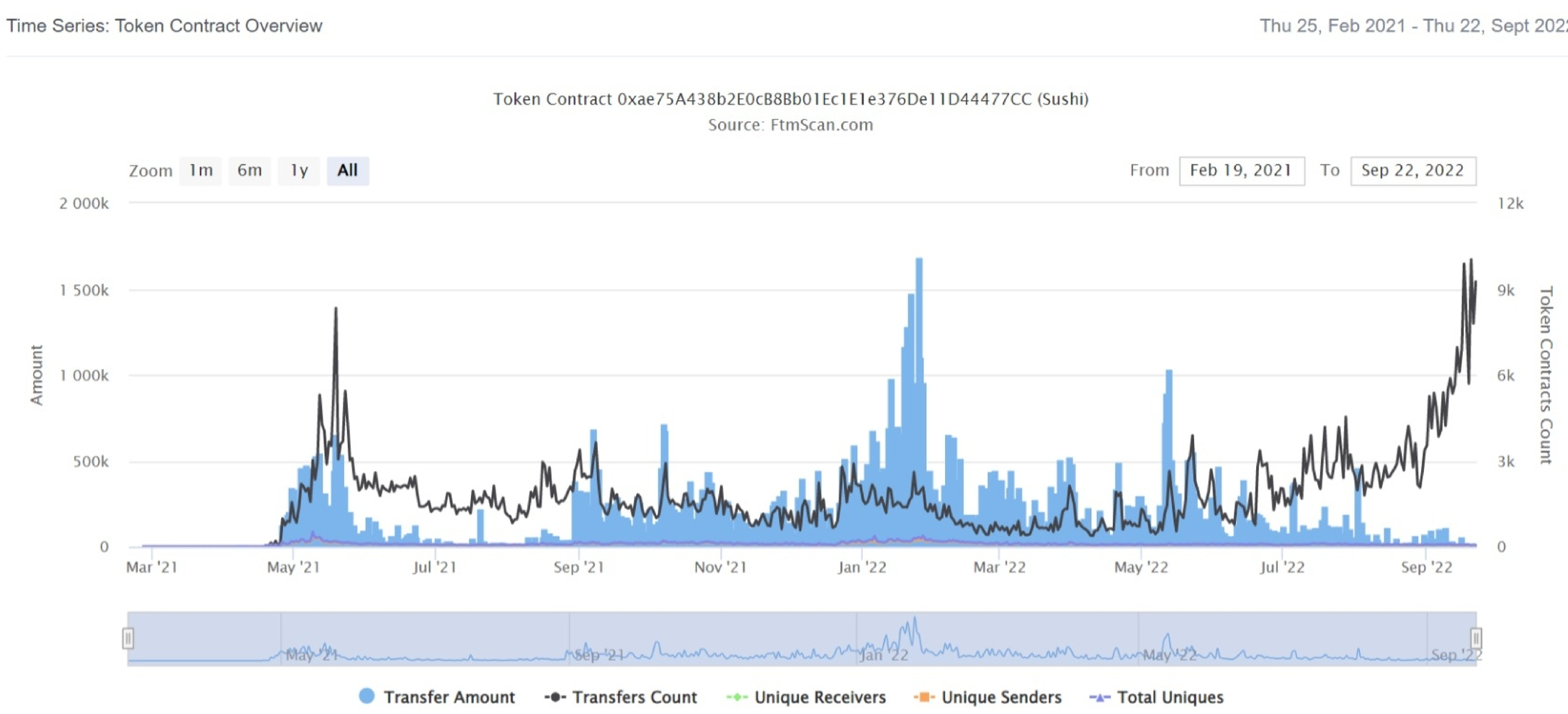
Sushi Top 20 Token Holders

(A total of 608,561.19 tokens held by the top 100 accounts from the total supply of 622,663.94 token)

Rank	Address	Quantity (Token)	Percentage
1	0xf84e313b36e86315af7a06ff26c8b20e9eb443c3	223,117.74837434279307352	35.8328%
2	0x525635374307559dd55c64f15cf101515259db41	87,384.215369684536183976	14.0339%
3	0x7d81ebcc3234125c1ebd5ceb82b1ba53c3a26887	68,921.737505979470933907	11.0689%
4	0x299e18a53ef3022891b68f27035e4246efd7c6f8	41,718.217490231690532491	6.7000%
5	0x75c8e2dd57927eb0373e8e201ebf582406adcf45	21,074.533512574280740042	3.3846%
6	0x49d2e0dc99c7358d7a9a8633bf6df111d0ee7f65	18,497.455219945670747889	2.9707%
7	0x5b9451b1bfae2a74d7b9d0d45bdd0e9a27f7bb22	11,021.870493516927740053	1.7701%
8	0x82f07a583724c80aaa109987f5949d14e2d836e7	10,733.437369609787172985	1.7238%
9	0xa90eca853a5acef06c287802e2632d9331fd3d2a	10,506.544251506698450439	1.6874%
10	0x39dd07ac211c33e796e6d89c4df9481e0b60ecfe	10,149.207590401453822124	1.6300%
11	0xdda77165dba487917e82b96d013b0895fb36c9e4	9,959.465203813056689188	1.5995%
12	0x5fb8a880686317c4435be72d62b57576c53173c2	9,791.137001629488328821	1.5725%
13	Iron Bank: iSUSHI Token	7,490.30587237710932058	1.2029%
14	0x4cd80aa0ce4881eb8679eda1f6fbe3d89aec0f7f	7,454.148591114470612633	1.1971%
15	0x5b84fd97de436bb64a866ec122985cf992d535bc	6,395.994523684743745955	1.0272%
16	0x9fe4c0ce5f533e96c2b72d852f190961ad5a7bb3	6,213.506188231377242027	0.9979%
17	Beethoven X: Vault	5,993.634783097354772376	0.9626%
18	0xf731202a3cf7efa9368c2d7bd613926f7a144db5	3,653.225175527547192604	0.5867%
19	0x582c34536d7f4a2a5d96118d5d80ba3000c6b5af	3,268.3864	0.5249%
20	0x68f5da4a7027705590220143ae1073877076fff5	3,264.154641615021270872	0.5242%

Sushi Token Distribution

Sushi Contract overview



Contract functions details

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

+[Int] IERC2612
    -[Ext] permit
    -[Ext] nonces

+[Int] IWERC10 (IERC20, IERC2612)
    -[Ext] approveAndCall
    -[Ext] transferAndCall

+[Int] ITransferReceiver
    -[Ext] onTokenTransfer

+[Int] IApprovalReceiver
    -[Ext] onTokenApproval

+[Lib] Address
    -[Int] isContract

+[Lib] SafeERC20
    -[Int] safeTransfer
    -[Int] safeTransferFrom
    -[Int] safeApprove
    -[Pvt] callOptionalReturn

+Sushi (IWERC10)
    -[Pub] owner
    -[Pub] changeDCRMOwner #
      -modifiers: onlyOwner
    -[Pub] Swapin #
      -modifiers: onlyOwner
    -[Pub] Swapout #
    - <constructor>
    -[Ext] totalSupply

Contract functions details

-[Int] _mint #
-[Int] _burn #
-[Ext] approve #
-[Ext] approveAndCall #
-[Ext] permit #
-[Ext] transferWithPermit #
-[Int] verifyEIP712
-[Int] verifyPersonalSign
-[Int] prefixed
-[Ext] transfer #
-[Ext] transferFrom #
-[Ext] transferAndCall #

($) = payable function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
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.	High issue
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.	Passed
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
20.	Too old version	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

One high severity issues found.

1. Timestamp dependence.

- **Description**

  The contract have used block.timestamp many times in some function such as changeDCRMOwner, permit, transferWithPermit, as the miners here can manipulate the smart contract in order to attack the contract.

- **Recommendation**

  We advise you to not use block.timestamp in your contract and apply the 15-second rule which says that If the scale of your time-dependent event can vary by 15 seconds and maintain integrity, it is safe to use a block.timestamp

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

No low severity issues found.

Centralization

Owner Privileges :

- Sushi Contract:
    - Owner can change DCRM Owner.
    - Owner can swapin.

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble as smart contract ownership has not been renounced. Following are Admin functions functions:

- Swapin
- Changedcrmowner

Conclusion

Smart contract contains high 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.