Smart Contract Security Audit Report

Symbiosis Metarouter OnchainSwap

1. Contents

2. General Information

This report contains information about the results of the security audit of the Symbiosis (hereafter referred to as "Customer") smart contracts, conducted by Decurity in the period from 11/09/2023 to 18/09/2023.

2.1. Introduction

Tasks solved during the work are:

- Review the protocol design and the usage of 3rd party dependencies,
- Audit the contracts implementation,
- Develop the recommendations and suggestions to improve the security of the contracts.

2.2. Scope of Work

The audit scope included the following contracts:

- https://github.com/symbiosis-finance/core-contracts/blob/main/contracts/periphery/OnchainGateway.sol
- https://github.com/symbiosis-finance/core-contracts/blob/main/contracts/periphery/OnchainSwapV3.sol

Initial review was done off Github and the rechecking was done for the commit https://github.com/symbiosis-finance/core-contracts/commit/8f475de5fd303c5a9e424e0010dc748cd5abe0c4.

2.3. Threat Model

The assessment presumes the actions of an intruder who might have the capabilities of any role (an external user, token owner, token service owner, or a contract). The risks of centralization were not taken into account at the Customer's request.

The main possible threat actors are:

- User,
- Protocol owner,
- Liquidity Token owner/contract.

The table below contains sample attacks that malicious attackers might carry out.

Table. Theoretically possible attacks

Attack	Actor
Contract code or data hijacking *Deploying a malicious contract or submitting malicious data*	Contract owner Token owner
Financial fraud *A malicious manipulation of the business logic and balances, such as a reentrancy attack or a flash loan attack*	Anyone
Attacks on implementation *Exploiting the weaknesses in the compiler or the runtime of the smart contracts*	Anyone

2.4. Weakness Scoring

An expert evaluation scores the findings in this report, and the impact of each vulnerability is calculated based on its ease of exploitation (based on the industry practice and our experience) and severity (for the considered threats).

2.5. Disclaimer

Due to the intrinsic nature of the software and vulnerabilities and the changing threat landscape, it cannot be generally guaranteed that a certain security property of a program holds.

Therefore, this report is provided “as is” and is not a guarantee that the analyzed system does not contain any other security weaknesses or vulnerabilities. Furthermore, this report is not an endorsement of the Customer’s project, nor is it an investment advice.

That being said, Decurity exercises the best effort to perform its contractual obligations and follow the industry methodologies to discover as many weaknesses as possible and maximize the audit coverage using limited resources.

3. Summary

As a result of this work, we have discovered a 1 medium security issue, which has been fixed.

The other suggestions included fixing the low-risk issues and some best practices (see Security Process Improvement).

3.1. Suggestions

The table below contains the discovered issues, their risk level, and their status as of May 3, 2023.

Table. Discovered weaknesses

Issue	Contract	Risk Level	Status
OnchainSwap DoS	contracts/periphery/OnchainSwapV3.sol	Medium	Fixed
Centralization risks	contracts/periphery/OnchainSwapV3.sol	Low	Acknowledged

4. General Recommendations

This section contains general recommendations on how to improve the overall security level.

The Findings section contains technical recommendations for each discovered issue.

4.1. Security Process Improvement

The following is a brief long-term action plan to mitigate further weaknesses and bring the product security to a higher level:

- Keep the whitepaper and documentation updated to make it consistent with the implementation and the intended use cases of the system,
- Perform regular audits for all the new contracts and updates,
- Ensure the secure off-chain storage and processing of the credentials (e.g. the privileged private keys),
- Launch a public bug bounty campaign for the contracts.

5. Findings

5.1. OnchainSwap Contract DoS

Risk Level: Medium

Status: Fixed (custom `forceApprove` implementation introduced)

Contracts:

- contracts/periphery/OnchainSwapV3.sol

Location: Lines: 47. Function: onswap.

Description:

The `OnchainSwapV3` contract approves tokens to the DEX without checking for an existing approval. For certain `ERC20` tokens, such as USDT, in order to change the approved amount, you must first reduce the allowance for the address to zero by calling `approve(_spender, 0)`.

Users can break any interaction with such token by approving `1 wei` of the token to DEX through the `onswap()` function. They need to provide appropriate `calldata_` for approve and token address as dex argument.

```
contracts/periphery/OnchainSwapV3.sol:67

(bool swapPassed, ) = dex.call{value: msg.value - fee}(
            calldata_
        );
```

After that all subsequent attempts to approve this token to the DEX from `OnchainSwapV3` contract will be reverted because the allowance is already non-zero.

Remediation:

Consider using the `forceApprove()` function from the OpenZeppelin repository. If a revert occurs, it will call `approve(_spender, 0)` and attempt the operation again.

References:

- https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/utils/SafeERC20.sol#L76

5.2. Centralization risks

Risk Level: Low

Status: Acknowledged

Description:

`OnchainSwapV3` contract is controlled by the EOA `owner` account. To reduce centralization risks we suggest several improvements:

1) Admin function `changeFee()` from `OnchainSwapV3` contract does not validate its argument `_newFee`. To reduce risks and protect users from losing money, it is better to check that the value is not too high.
2) For emergency cases you can use `pause()` from `Pausable` library instead of changing fees to extremely high values.
3) Timelock contract can become the `owner`.

Remediation:

Consider implementing some of the recommendations to decrease centralization risks.

6. Appendix

6.1. About us

The Decurity team consists of experienced hackers who have been doing application security assessments and penetration testing for over a decade.

During the recent years, we've gained expertise in the blockchain field and have conducted numerous audits for both centralized and decentralized projects: exchanges, protocols, and blockchain nodes.

Our efforts have helped to protect hundreds of millions of dollars and make web3 a safer place.