Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

December, 2021

Audit Details

Audited project

Senpai Vault

Deployer address

0x00f596aD5f6c5a0C666991260E60aa1EBD06B619

Client contacts:

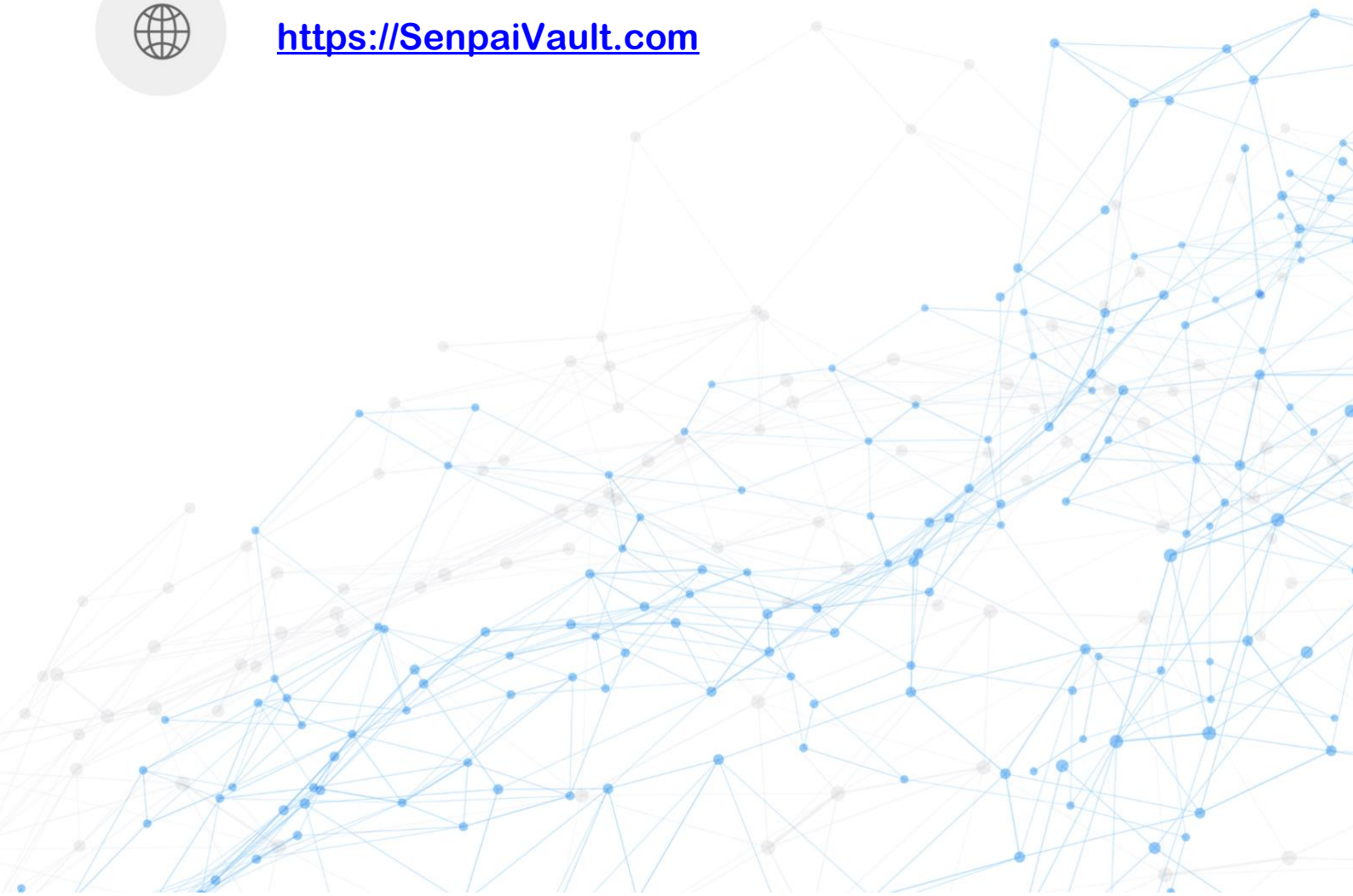
Senpai Vault team

Blockchain

Ethereum

Project website:

https://SenpaiVault.com



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 Senpai Vault to perform an audit of smart contracts:

https://etherscan.io/address/0x9514ed01ba8790799d67879e4cc773ffee96c885#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

Contract details for 10.12.2021

Contract name	UniswapV2Locker
Contract address	0x9514eD01Ba8790799D67879E4cC773FfEE96c885
Contract deployer address	0x00f596aD5f6c5a0C666991260E60aa1EBD06B619
Contract's current owner address	0x060b40a06803aEE172958d769d4a7e95b5E9a58b

Contract functions details

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

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

+ [Lib] TransferHelper
  - [Int] safeApprove #
  - [Int] safeTransfer #
  - [Int] safeTransferFrom #

+ ReentrancyGuard
  - [Int] <Constructor> #

+ [Lib] EnumerableSet
  - [Prv] _add #
  - [Prv] _remove #
  - [Prv] _contains
  - [Prv] _length
  - [Prv] _at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at

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

+ [Int] IUniswapV2Pair
  - [Ext] factory
  - [Ext] token0
  - [Ext] token1

+ [Int] IERCBurn
  - [Ext] burn #
  - [Ext] approve #
  - [Ext] allowance #
  - [Ext] balanceOf

+ [Int] IUniFactory
  - [Ext] getPair

+ [Int] IMigrator
  - [Ext] migrate #

+ UniswapV2Locker (Ownable, ReentrancyGuard)
  - [Pub] <Constructor> #
  - [Pub] setDev #
    - modifiers: onlyOwner
  - [Pub] setMigrator #
    - modifiers: onlyOwner
  - [Pub] setSecondaryFeeToken #
    - modifiers: onlyOwner
  - [Pub] setReferralTokenAndHold #
    - modifiers: onlyOwner
  - [Pub] setFees #
    - modifiers: onlyOwner
  - [Pub] whitelistFeeAccount #
    - modifiers: onlyOwner
  - [Ext] lockLPToken ($)
    - modifiers: nonReentrant
  - [Ext] relock #
    - modifiers: nonReentrant
  - [Ext] withdraw #
    - modifiers: nonReentrant
  - [Ext] incrementLock #
    - modifiers: nonReentrant
  - [Ext] splitLock ($)
    - modifiers: nonReentrant
  - [Ext] transferLockOwnership #
  - [Ext] migrate #
    - modifiers: nonReentrant
  - [Ext] getNumLocksForToken
  - [Ext] getNumLockedTokens
  - [Ext] getLockedTokenAtIndex
  - [Ext] getUserNumLockedTokens
  - [Ext] getUserLockedTokenAtIndex
  - [Ext] getUserNumLocksForToken
  - [Ext] getUserLockForTokenAtIndex
  - [Ext] getWhitelistedUsersLength
  - [Ext] getWhitelistedUserAtIndex
  - [Ext] getUserWhitelistStatus

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 change dev address.

```
function setDev(address payable _devaddr) public onlyOwner {
  devaddr = _devaddr;
}
```

- Owner can change migrator.

```
function setMigrator(IMigrator _migrator) public onlyOwner {
  migrator = _migrator;
}
```

- Owner can change secondary fee token.

```
function setSecondaryFeeToken(address _secondaryFeeToken) public onlyOwner {
  gFees.secondaryFeeToken = IERCBurn(_secondaryFeeToken);
}
```

- Owner can change referral token and hold amount.

```
function setReferralTokenAndHold(IERCBurn _referralToken, uint256 _hold) public onlyOwner {
  gFees.referralToken = _referralToken;
  gFees.referralHold = _hold;
}
```

- Owner can change fees.

```
  function setFees(
      uint256 _referralPercent,
      uint256 _referralDiscount,
      uint256 _ethFee,
      uint256 _secondaryTokenFee,
      uint256 _secondaryTokenDiscount,
      uint256 _liquidityFee
  ) public onlyOwner {
      gFees.referralPercent = _referralPercent;
      gFees.referralDiscount = _referralDiscount;
      gFees.ethFee = _ethFee;
      gFees.secondaryTokenFee = _secondaryTokenFee;
      gFees.secondaryTokenDiscount = _secondaryTokenDiscount;
      gFees.liquidityFee = _liquidityFee;
  }
```

- Owner can add or remove address from whitelist.

```
function whitelistFeeAccount(address _user, bool _add) public onlyOwner {
  if (_add) {
    feeWhitelist.add(_user);
  } else {
    feeWhitelist.remove(_user);
  }
}
```

Conclusion

Smart contracts do not contain high severity issues! Smart contracts contain owner privileges. The further transfers and operations with the funds raise are not related to this particular contract.

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.