Security Assessment

SwapSicle

Verified by Vibranium Audits on 03 October 2023

Revised on 05 October 2023

Vibranium Audits Verified on October 3rd, 2023 / Revised on October 5th, 2023

Swapsicle

The security assessment was prepared by Vibranium Audits.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DEFI	Ethereum	Manual Review & Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 03/10/2023 Revised on 05/10/2023	N/A

CODEBASE	COMMITS
https://github.com/swapsicledex/swapsicle-v2-contracts/tree/master	b474271943210921947b09bc6e005d55c8316db8

Vulnerability Summary

29	29	0	0	29	0	29
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

1	Critical		Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation by external or internal actors.
1	High	0 Resolved	High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation by external or internal actors.
1	Medium	0 Resolved	Medium vulnerabilities are usually limited to state manipulations, but cannot lead to assets loss. Major deviations from best practices are also in this category.
1	Low	0 Resolved	Low vulnerabilities are related to outdated and unused code or minor gas optimization. These issues won't have a significant impact on code execution, but affect the code quality.
0	Informational	0 Resolved	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | PULSEMARKETS

CODEBASE | SWAPSICLE

Repository

https://github.com/swapsicledex/swapsicle-v2-contracts/tree/master

Commit

b4742719432109219 47b09bc6e005d55c8316db8

AUDIT SCOPE | SWAPSICLE

5 files audited ● 5 files with Acknowledged findings ● 5 files with Resolved findings

ID	Files	Commit Hash
PCA	Vulnerabilities under PopsConverter.sol	b474271943210921947b09bc6e005d55c8316db8
ITA	Vulnerabilities under IceToken.sol	b474271943210921947b09bc6e005d55c8316db8
STA	Vulnerabilities under SlushToken.sol	b474271943210921947b09bc6e005d55c8316db8
IVA	Vulnerabilities under IceCreamVan.sol	b474271943210921947b09bc6e005d55c8316db8
ZVA	Vulnerabilities Under ZombieVan.sol	b474271943210921947b09bc6e005d55c8316db8
GSA	Vulnerabilities Found Globally	b474271943210921947b09bc6e005d55c8316db8

APPROACH & METHODS | SWAPSICLE

This report has been prepared for SWAPSICLE(2023) to discover issues and vulnerabilities in the source code of the SWAPSICLE project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the code base to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire code base by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices.
We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors.
- Enhance general coding practices for better structures of source codes.
- Review unit tests to cover the possible use cases.
- Review functions for readability, especially for future development work.

FINDINGS | SWAPSICLE

21	12	3	2	4	0
Total Findings	Critical	High	Medium	Low	Informational

This report has been prepared to discover issues and vulnerabilities for SWAPSICLE. Through this audit, we have uncovered 29 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
ZVA-01	Reentrancy-eth	Logical Issue	High	Acknowledged
ZVA-02	Unchecked Transfers	Logical Issue	Critical	Acknowledged
GSA-01	Floating Pragma Usage	Logical Issue	Minor	Acknowledged
PCA-01	Unchecked Approvals	Logical Issue	Medium	Acknowledged
PCA-02	Unchecked Transfers	Logical Issue	Critical	Acknowledged
IVA-01	Unchecked Transfers	Logical Issue	Critical	Acknowledged
STA-01	Unchecked Transfers	Logical Issue	Critical	Acknowledged

ZVA-01 | Reentrancy-eth

Categor y	Severity	Location	Status
Logical Issue	High	ZombieVan.sol	Revised

Description (3 findings)

A reentrancy is a programmatic approach in which an attacker performs recursive withdrawals to steal all Ethers locked in a contract.

Although the 'unstake(..)' function logically implements some measures against reentrancy by setting state variables before making transfer calls etc...
However it is still potentially viable to have a Reentrancy attack happen through :

- stakeToken.transfer(msg.sender, userAmount);
- _claim()
- claim()

```
// Transfer the staked tokens back to the user (and fee to feeAddress)
uint256 userAmount = amount - fee;
stakeToken.transfer(msg.sender, userAmount);

// Update the user's deposit balance and total withdrawals
userDeposits[msg.sender] -= amount;
totalAccumulatedWithdrawals += amount;
totalStaked -= amount;

// Call the claim function to transfer the rewards
_claim();
```

Recommendation

Just like in the other smart contracts like 'IceCreamVan.sol', implement Openzeppelin's ReentrancyGuard.sol smart contract with the 'nonReentrant' modifier on the 'withdraw()' function (not needed on _claim() because it's private and it doesn't call withdraw() back.)

ZVA-02 PCA-02 IVA-01 SRA-01 | Unchecked Transfers

Categor y	Severity	Location	Status
Logical Issue	Critical	Global, but mostly due to the nature of SlushToken.sol	Revised

Description (12 Findings)

The return value of an external transfer/transferFrom call is not checked, while the tokens do not revert in case of failure and return false. Any deposit or transfer in general will not revert if the transfer fails, and an attacker can call deposit for free or a user could incur losses.

This is mostly due to the SlushToken being based on OFTV2, which in itself is based on ERC20 which does not protect against this potential problem.

Therefore it is recommended to check the return value of transfers before proceeding with any state changes.

Presence in Codebase:

- **IceCreamVan.sweepErc20(IERC20)**
- **IceToken._convert(uint256,address)**
- **IceToken._finalizeRedeem(address,uint256,uint256,uint256) (L400)**
- **IceToken._finalizeRedeem(address,uint256,uint256,uint256) (L418)**
- **PopsConverter.convert(uint256,uint8)**
- **PopsConverter.withdrawToken(IERC20,uint256)**
- **PopsConverter._prepareTransfer(uint256,uint8,uint16)**

Recommendation

As IceToken.sol has ‘SafeERC20’ implemented, you can either override OFTV2's codebase to implement SafeERC20 and use ‘safeTransfer’/’safeTransferFrom’, or hardcode the transfer checks with every transfer call.

Revision

Added necessary checks to ensure that transfers/transferFroms succeeded before proceeding.

GSA-01 | Floating Pragma

Categor y	Severity	Location	Status
Logical Issue	Minor	Global Scope	Revised

Description (4 findings)

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Recommendation

Use a new specific stable Solidity version e.g 0.8.18, although avoid using the latest version avoiding any potential yet undiscovered bugs.

Revision

The SwapSicle team decided to move forward with implement Solidity 0.8.19 for their smart contracts.

PCA-01 | Unchecked Approvals

Categor y	Severity	Location	Status
Logical Issue	Medium	PopsConverter.sol	Revised

Description (2 Findings)

Similar to the Unchecked Transfers issue, it is needed to check the return value of approve functions to ensure that users do not incur any potential losses.

```
function convert(uint256 popsAmount, uint8 tokenId) external nonReentrant {
    if (tokenId > 1) revert InvalidTokenId();
    if (!conversionAllowed[0][tokenId]) revert ConversionNotAllowed();
    _validatePops(popsAmount);

    (, uint256 targetAmount) = _prepareTransfer(popsAmount, tokenId, 0);
    if (tokenId == 0) {
        slush.transfer(msg.sender, targetAmount); // send SLUSH to the user
    } else {
        slush.approve(address(ice), targetAmount);
        ice.convertTo(targetAmount, msg.sender); // convert SLUSH to ICE for the user
    }

    emit Convert(msg.sender, popsAmount, targetAmount, tokenId);
}
```

```
function crossChainConvert(uint256 popsAmount, uint8 tokenId, uint16 chainId
) external payable nonReentrant validateConversion(tokenId, chainId) {
    _validatePops(popsAmount);

    uint16 lzChainId = chainIdToLzChainId[chainId];
    (address tokenAddress, uint256 targetAmount) = _prepareTransfer(popsAmount, tokenId, chainId);
    if (tokenId == 1) {
        slush.approve(address(ice), targetAmount);
        ice.convert(targetAmount); // convert SLUSH to ICE
    }
    IOFTV2(tokenAddress).sendFrom{value: msg.value}(
        address(this), lzChainId, LzLib.addressToBytes32(msg.sender), targetAmount,
        ICommonOFT.LzCallParams(payable(msg.sender), address(0), "")
    );

    emit CrossChainConvert(msg.sender, popsAmount, targetAmount, tokenId, chainId);
}
```

Recommendation

Same Recommendation as PCA-02.

Recommendation

Added necessary checks to make sure Approvals are passed before proceeding.

DISCLAIMER | VIBRANIUM AUDITS

This report is subject to the terms and conditions (including without limitation description of services confidentiality disclaimer and limitation of liability) set forth in the Services Agreement or the scope of services and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement This report may not be transmitted disclosed referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Vibranium Audits prior written consent in each instance

This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team This report is not nor should be considered an indication of the economics or value of any “product” or “asset”created by any team or project that contracts Vibranium Audits to perform a security assessment This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed nor do they provide any indication of the technologies proprietors business business model or legal compliance

This report should not be used in any way to make decisions around investment or involvement with any particular project This report in no way provides investment advice nor should be leveraged as investment advice of any sort This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology

Blockchain technology and cryptographic assets present a high level of ongoing risk Vibranium Audits position is that each company and individual are responsible for their own due diligence and continuous security Vibranium Audits goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze

The assessment services provided by Vibranium Audits is subject to dependencies and under continuing development You Agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty The assessment reports could include false positives false negatives and other unpredictable results The services may access and depend upon multiple layers of third-parties

ALL SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW VIBRANIUM AUDITS HEREBY DISCLAIMS ALL WARRANTIES WHETHER EXPRESS IMPLIED STATUTORY OR OTHERWISE WITH RESPECT TO THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE TITLE AND NON-INFRINGEMENT AND ALL WARRANTIES ARISING FROM COURSE OF DEALING USAGE OR TRADE PRACTICE WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF WILL MEET CUSTOMER’S OR ANOTHER PERSON’S REQUIREMENTS ACHIEVE ANY INTENDED RESULT BE COMPATIBLE OR WORK WITH ANY SOFTWARE SYSTEM OR OTHER SERVICES OR BE SECURE ACCURATE COMPLETE FREE OF HARMFUL CODE

OR ERROR-FREE WITHOUT LIMITATION TO THE FORGOING, VIBRANIUM AUDITS PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KINDTHAT WILL MEET CUSTOMER' S REQUIREMENTS ACHIEVE ANY INTENDED RESULTS BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE APPLICATIONS SYSTEMS OR SERVICES OPERATE WITHOUT INTERRUPTION MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED

WITHOUT LIMITING THE FOREGOING NEITHER VIBRANIUM AUDITS NOR ANY OF VIBRANIUM AUDITS AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND EXPRESS OR IMPLIED AS TO THE ACCURACY RELIABILITY ORCURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE VIBRANIUM AUDITS WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS MISTAKES OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE OF ANY NATURE WHATSOEVER RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS

ALL THIRD-PARTY MATERIALS ARE PROVIDED “AS IS” AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNEROR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS

THE SERVICES ASSESSMENT REPORT AND ANY OTHER MATERIALS HERE UNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT NOR MAY COPIES BE DELIVERED TO ANY OTHER PERSON WITHOUT VIBRANIUM AUDITS PRIOR WRITTEN CONSENT IN EACH INSTANCE

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS

THE REPRESENTATIONS AND WARRANTIES OF VIBRANIUM AUDITS CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER ACCORDINGLY NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE

FOR AVOIDANCE OF DOUBT THE SERVICES INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS ORMATERIALS SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Vibranium | Securing the Web3 World

Vibranium Audits is a blockchain security company that was founded in 2021 by professors from the University of Greenwich and cyber-security engineers from ITI Capital. As pioneers in the field, Vibranium Audits utilizes best-in-class Formal Verification and AI technology to secure and monitor blockchains, smart contracts, and Web3 apps.

