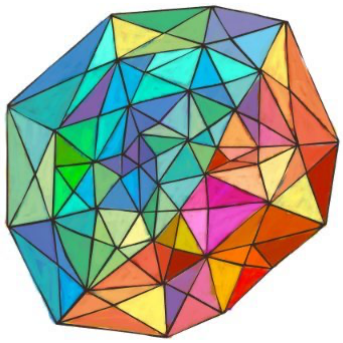
verichains

SECURITY AUDIT OF

KALEIDOSWAP SMART CONTRACT



Public Report

Feb 06, 2024

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
XAI	Xai is a permissionless Orbit chain leveraging the Arbitrum Nitro technology stack to bring an innovative, gaming-focused Layer 3 to the games industry
Solc	A compiler for Solidity.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Feb 06, 2024. We would like to thank the KaleidoCube for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the KaleidoSwap Smart Contract. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified one vulnerable issue in the application.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About KaleidoSwap Smart Contract

The vision of KaleidoCube is a commitment to the Arbitrum ecosystem, their foundational home. They believe in the power and potential of the Arbitrum ecosystem and its decentralized governance model. A portion of the KaleidoCube treasury will be converted to $ARB, accumulating a substantial stake in the ARB DAO. This strategic alignment not only strengthens their position within the ecosystem but also allows them to actively participate and contribute to the governance and future direction of Arbitrum.

KaleidoSwap is the forefront Automated Market Maker (AMM) Decentralized Exchange (DEX) in the KaleidoCube ecosystem. Forked from the reputable UniSwap V2, KaleidoSwap offers an advanced, user-friendly, and secure platform for seamless token exchanges on the XAI Blockchain.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the KaleidoSwap Smart Contract.

The latest version of the following files were made available in the course of the review:

SHA256 Sum	File
20a8ab1a110567a75415fe09a8521fdb061f3e35f50b36aeb99af3235427052b	KaleidoSwapFactory.sol
ec83c8d09072f287c10dbd57cf9fe1572be02d6d3815b8a85426fbcea08b38fa	KaleidoSwapPair.sol
c8a160540894cb96b8f61be35a889f41b5fd66e77921f29c7e702789840a4514	KaleidoSwapRouter.sol
d4b5c8370b61e848fcafd09f6d06c47c1a5900616092650654255 1c62db5d141	libraries/KaleidoSwapLibrary.sol
3540727e47ef742fd2c0c6938228229985c6d0087ec15fe654d462ca8048e2a1	libraries/UQ112x112.sol
1753dea01c9f1a2e30744a21115badd247fd0a39e8e48bab0c557975de0e54a9	WXAI.sol

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

KaleidoCube acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. KaleidoCube understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, KaleidoCube agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the KaleidoCube will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the KaleidoCube, the final report will be considered fully accepted by the KaleidoCube without the signature.

2. AUDIT RESULT

2.1. Overview

The KaleidoSwap Smart Contract was written in `Solidity` language, with the required version to be `^0.8.2`. The source code was written based on OpenZeppelin's library and UniswapV2.

The KaleidoSwap is a minor custom version of the UniswapV2. The platform is designed to provide a decentralized exchange for the KaleidoCube ecosystem. Some minor changes were made to the original UniswapV2 code to fit the requirements of the KaleidoCube ecosystem. We will show some minor changes in the following sections.

2.1.1. KaleidoSwapFactory

The Factory is responsible for creating the `KaleidoSwapPair` contract, with each pair representing a unique token pairing. Unlike the original `UniswapV2`, the `KaleidoSwapFactory` is an upgradable contract, providing deployers with the flexibility to alter its implementation.

2.1.2. KaleidoSwapPair

This contract encompasses essential functionalities required for executing token swaps, mirroring those found in `UniswapV2Pair`.

2.1.3. KaleidoSwapRouter

The Router module handles the execution of trades and routes transactions to the appropriate `tokens` pair. It provides functions for swapping tokens, adding liquidity to pairs, and removing liquidity. The `KaleidoSwapRouter` is also an upgradable contract and have the pause and unpause function. The `ADMIN_ROLE` can use these functions to pause and unpause the router.

2.1.4. WXAI

WXAI is a wrapped version of XAI, operating as a simple ERC20 token.

2.1.5. KaleidoSwapLibrary

The library is utilized for mathematical calculations. The `KaleidoSwapLibrary` is a slightly customized version of the `UniswapV2Library`, where `UniswapV2` error messages have been renamed to `Kaleido` error messages.

2.2. Findings

During the audit process, the audit team found 1 vulnerability in the given version of KaleidoSwap Smart Contract.

2.2.1. Upgradeable contract CRITICAL

Within the audit scope, the `KaleidoSwapFactory` and `KaleidoSwapRouter` are identified as upgradeable contracts, allowing the deployer to modify their logic. If the deployer account is compromised, attackers may exploit this feature to their advantage.

For instance, the `KaleidoSwapFactory` is responsible for creating the `KaleidoSwapPair` contract, where each pair represents a unique token pairing. If an attacker can manipulate the logic of the `KaleidoSwapFactory`, they could potentially reinitialize pair contracts with malicious tokens, thereby withdrawing significant amounts of valuable tokens from these pairs.

Similarly, with the `KaleidoSwapRouter`, attackers could alter the logic to engage in phishing attempts during the swap process.

RECOMMENDATION

We suggest changing all `upgradeable` abstract contract to normal contract.

UPDATES

- *Feb 06, 2024*: This issue has been acknowledged and fixed by the KaleidoCube team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Feb 03, 2024*	Private Report	Verichains Lab
1.1	*Feb 06, 2024*	Public Report	Verichains Lab

Table 2. Report versions history