SOLIDITY FINANCE

KALIDAO

Smart Contract Audit Report

AUDIT SUMMARY

KALIDAO is an new lightweight implementation of a multi-sig wallet where signers are determined by NFTs.

For this audit, we reviewed the ClubNFT, ClubLoot, KaliClubSig, and KaliClubSigFactory contracts at commit 5923a2ccd98016c288bd1e174139049bc23d531e on the team's GitHub repository.

AUDIT FINDINGS

> *Please ensure trust in the team prior to investing.*
>
> *Date: April 5th, 2022.*

24db237a4655f7cbd973cb565ac639cfadfeac73 to commit 5923a2ccd98016c288bd1e174139049bc23d531e.

Finding #1 - ClubSig - Informational (Acknowledged)

Description: *The execute() function runs until a quorum or invalid signature is reached.*
Risk/Impact: *This will lead to excess gas costs in specific situations.*
Recommendation: *The team should consider having the function initially revert if sufficient signatures are not provided.*
Resolution: *The team has acknowledged this behavior and chosen to not implement the recommendation.*

Finding #2 - ClubSig - Informational (Resolved)

Description: *The init() and govern() functions allow the quorum to be 0.*
Risk/Impact: *The execute() function will execute any arbitrary logic if the quorum is 0.*
Recommendation: *The team should consider preventing a quorum of 0.*
Resolution: *The team has implemented a check to prevent a quorum of 0.*

CONTRACTS OVERVIEW

> - *The contracts are implemented in a highly gas efficient manner leveraging assembly and unchecked math where applicable.*
>
> *KaliClubSig Contract:*
>
> - *This contract serves as an NFT based multi-sig wallet for EIP-712 and EIP-1271 signed transactions.*
> - *When the contract is initialized, a specified number of addresses that serve as valid signers are minted an NFT; the signers must be passed in ascending order by address or the initialization will fail.*
> - *A separate ClubLoot contract is used to track users' "loot" balance.*
> - *A non-zero quorum amount, base URI, and "redemption start" time is also specified.*
> - *The contract may optionally be initialized as paused, disabling NFT transfers.*
> - *NFT metadata that contains information about the NFTs is stored using an off-chain URI endpoint.*
> - *For non-Governors to execute a transaction, a quorum of signers must provide a signed message of the proposed transaction; each signer must own an NFT to be a valid signer.*
> - *If the signer is a contract, the contract must return whether the*

> - *The signatures must be provided in ascending order by signer address or the transaction will fail.*
> - *If a quorum is reached, the transaction is executed by the contract; the transaction may also optionally be executed as a delegated call.*
> - *Once the redemption period has passed, users may perform a "rage quit". A rage quit will burn a user's loot by the specified amount and provide the user with a proportional amount of all of the provided assets; the assets must be provided in ascending order by address or the transaction will fail.*
> - *Any Governor address may execute a transaction without requiring a quorum.*
> - *Any Governor or the contract itself may add or remove an address as a Governor at any time.*
> - *Any Governor or the contract itself may mint or burn NFTs from any address at any time; the quorum may also be updated to a non-zero value.*
> - *Any Governor or the contract itself may mint loot to any address at any time; the contract must be given the ability to mint in the ClubLoot contract or this will fail.*
> - *Any Governor or the contract itself may pause and unpause the contract and the ClubLoot contract at any time.*
> - *Any Governor or the contract itself may update the base URI and document data at any time.*

> *ClubLoot Contract:*
>
> - *This contract is used to track users' loot balance in the form of a minimal implementation of an ERC20 token.*
> - *When the contract is initialized, a list of Club members and loot amounts are specified where each member is subsequently minted the specified amount of loot.*
> - *The Governance address is specified upon deployment. The contract may also optionally be paused upon deployment.*
> - *Any user may burn their tokens to reduce the total supply.*
> - *Any user may burn tokens on another user's behalf if an allowance has been granted.*
> - *Each loot additionally represents votes intended to be used in a DAO where one loot represents one vote.*
> - *Users may delegate their votes to another address allowing them to vote on behalf of the user.*
> - *Once votes are delegated, the user must explicitly delegate back to themselves to regain their votes.*
> - *Users have the option to "permit" through the use of a signed message, allowing for a gasless approval by the user.*
> - *The Governance address may mint loot to and burn loot from any address at any time.*
> - *The Governance address may update its own address at any time.*
> - *The Governance address may pause the contract, disabling*

> *ClubSigFactory Contract:*
>
> - *Any user may use this contract to deploy clones of a master copy of the KaliClubSig and ClubLoot contracts.*
> - *A name, symbol, quorum, redemption start time, list of valid signers, and whether the contract is initially paused must be provided when cloning the contract.*
> - *The provided parameters will be used to initialize the cloned contracts.*
> - *If a "docs" string is provided a RicardianLLC will be minted; the RicardianLLC contract is out of the scope for this audit so we are unable to provide an assessment with regard to security.*

AUDIT RESULTS

Vulnerability Category	**Notes**	**Result**
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	N/A	PASS

Vulnerability Category	Notes	Result
Compiler Issues	N/A	PASS
Delegate Call to Untrusted Contract	N/A	PASS
Dependence on Predictable Variables	N/A	PASS
Ether/Token Theft	N/A	PASS
Flash Loans	N/A	PASS
Front Running	N/A	PASS
Improper Events	N/A	PASS
Improper Authorization Scheme	N/A	PASS

Vulnerability Category	Notes	Result
Logical Issues	N/A	PASS
Oracle Issues	N/A	PASS
Outdated Compiler Version	N/A	PASS
Race Conditions	N/A	PASS
Reentrancy	N/A	PASS
Signature Issues	N/A	PASS
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

INHERITANCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

ClubLoot Contract

INHERITANCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

KaliClubSigFactory Contract

INHERITANCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

ABOUT SOLIDITY FINANCE

Solidity Finance was founded in 2020 and quickly grew to have one of the most experienced and well-equipped smart contract auditing teams in the industry. Our team has conducted 1000+ solidity smart contract audits covering all major project types and protocols, securing a total of over $10 billion U.S. dollars in on-chain value.
Our firm is well-reputed in the community and is trusted as a top smart contract auditing company for the review of solidity code, no matter how complex. Our team of experienced solidity smart contract auditors performs audits for tokens, NFTs, crowdsales, marketplaces, gambling games, financial protocols, and more!

Contact us today to get a free quote for a smart contract audit of your project!

WHAT IS A SOLIDITY AUDIT?

Typically, a smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. A *Solidity Audit* takes this a step further by verifying economic logic to ensure the stability of smart contracts and highlighting privileged functionality to create a report that is easy to understand for developers and community members alike.

HOW DO I INTERPRET THE FINDINGS?

Each of our Findings will be labeled with a Severity level. We always recommend the team resolve High, Medium, and Low severity findings prior to deploying the code to the mainnet. Here is a breakdown on what each Severity level means for the project:

which can prevent the code from operating as intended.

- **Medium** severity issues are those which place at least some users' funds at risk and has a medium to high probability of exploitation.
- **Low** severity issues have a relatively minor risk association; these issues have a low probability of occurring or may have a minimal impact.
- **Informational** issues pose no immediate risk, but inform the project team of opportunities for gas optimizations and following smart contract security best practices.

GO HOME

Please note we are not associated with the Solidity programming language or the core team which develops the language.