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, ClubSig, and ClubSigFactory contracts at commit 24db237a4655f7cbd973cb565ac639cfadfeac73 on the team's GitHub repository.

AUDIT FINDINGS

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

> ***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.*
>
> ## Finding #2 - ClubSig - Informational
>
> ***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.*

CONTRACTS OVERVIEW

> - *The contracts are implemented in a highly gas efficient manner leveraging assembly where applicable.*
>
> *ClubSig Contract:*
>
> - *This contract serves as an NFT based multi-sig wallet for EIP-712*

> - *When the contract is initialized, a specified number of addresses that serve as valid signers are minted an NFT and allocated a "loot amount".*
> - *The signers must be passed in ascending order by address or the initialization will fail.*
> - *A quorum amount and base URI 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.*
> - *In order to execute a transaction, a quorum number of signers must provide a signed message of the proposed transaction; each signer must own an NFT in order to be a valid signer.*
> - *The signatures must be provided in ascending order by signer 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.*
> - *Users may also perform a "rage quit" at any time. 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.*
> - *Addresses may be designated as a Governor address by another*

> - *Any Governor address may pass an arbitrary transaction to the contract where it is subsequently executed; the transaction may also optionally be executed as a delegated call.*
> - *Any Governor or the contract itself may mint or burn NFTs from any address at any time; the quorum may also be updated.*
> - *Any Governor or the contract itself may pause and unpause the contract at any time.*
> - *Any Governor or the contract itself may update the base URI at any time.*
>
> *ClubSigFactory Contract:*
>
> - *Any user may use this contract to deploy clones of a master copy of the ClubSig contract.*
> - *A name, symbol, quorum, list of valid signers, and whether the contract is initially paused must be provided when cloning the contract.*

AUDIT RESULTS

Vulnerability Category	**Notes**	**Result**

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

Vulnerability Category	Notes	Result
Improper Authorization Scheme	N/A	PASS
Integer Over/Underflow	N/A	PASS
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

Vulnerability Category	Notes	Result
Overall Contract Safety		PASS

ClubSig Contract

INHERITANCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

ClubSigFactory Contract

INHERITANCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

Please review our Terms & Conditions and Privacy Policy. By using this site, you agree to these terms.

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:

- **High** severity indicates that the issue puts a large number of users' funds at risk and has a high probability of exploitation, or the smart contract contains serious logical issues 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

- **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.