Uno DAO

Smart Contract Audit Report

AUDIT SUMMARY

Uno DAO is a new platform in which users can participate in yield farming through the use of governance tokens.

For this audit, we reviewed the project team's VeUnoDaoYieldDistributor, Ownership, Owned, Resolver, SmartWalletChecker, Helpers, and TransferHelper contracts at commit 2332f79670f67530f71dbdad2f54949893c1fb0d on the team's GitHub repository.

AUDIT FINDINGS

Informational findings were identified and the team may want to address these issues. In addition, centralized aspects are present.
Date: December 29th, 2023

Finding #1 - SmartWalletChecker & Helpers - Informational

Description: *The* `SmartWalletChecker.check()` *and* `Helpers.isContract()` *functions do not properly determine if an address is a contract. These functions check if the address' code size is 0, but any code size checks made during a contract's construction will return a value of 0. This can allow contracts to bypass these functions by executing logic during their construction.*
Risk/Impact: *As these functions are not used anywhere within the scope of this audit, we are unable to determine their intended use along with potential impact.*
Recommendation: *These functions should not be used within a contract to determine if an address is a contract; they should be used for front end purposes only.*

Finding #2 - VeUnoDaoYieldDistributor - Informational

Description: *The first conditional block in the following code is redundant, as it is covered within the second conditional with the same resulting logic.*

```
if (
    storedEndTimestamp != 0 && (block.timestamp >= storedEndTimestamp)
) {
    eligibleVeUnoBal = 0;
} else if (block.timestamp >= storedEndTimestamp) {
    eligibleVeUnoBal = 0;
} else {
    eligibleVeUnoBal = currVeUnoBal;
}
```

Recommendation: *The first of the three conditional statements can be removed for gas savings on certain function calls.*

CONTRACTS OVERVIEW

- *The team must exercise caution when assigning the reward token to avoid using fee-on-transfer tokens unless the proper exemptions are made.*
- *As the contracts are implemented with Solidity v0.8.0, they are safe from any possible overflows/underflows.*

VeUnoDaoYieldDistributor Contract:

- *Any address included in the Reward Notifier list can deposit rewards into this contract at any time.*
- *If rewards are not currently being distributed, the deposited rewards are distributed linearly over the length of the contract's yield duration.*
- *When rewards are deposited during an existing distribution period, a new yield period begins using the unvested rewards combined with the newly deposited tokens.*

- *In order to begin to participate in yield distributions, a user must have a locked veUNO balance and must first be checkpointed within this contract.*
- *Any user can checkpoint themselves or another user at any time.*
- *Users are earned a portion of distributed rewards over the course of a yield period equal to their locked veUNO balance in relation to the total veUNO supply.*
- *A user's total pending yield is updated any time a new checkpoint is made, or if they harvest rewards.*
- *If a user's locked balance has increased at the time of a new checkpoint, their previous balance is used for pending reward calculations.*
- *If the user's locked balance has decreased, the average of their current and previous balances are used for reward calculations.*
- *As a result, users should checkpoint any time their balance increases to maximize their yield.*
- *Users only earn rewards while their tokens are locked.*
- *The owner or Timelock address can pause the contract at any time, preventing all yield claims.*
- *The owner or Timelock address can update the Timelock address at any time.*
- *The owner or Timelock address can manually update the yield rate to any amount at any time.*
- *The owner or Timelock address can update the yield duration to any length once the period finish has been reached.*
- *The owner or Timelock address can withdraw any token from the contract at any time.*
- *The owner or Timelock address can add or remove any address from the Greylist or Reward Notifier list at any time.*
- *The contract's owner is is specified upon deployment.*
- *The owner address can "nominate" a new owner at any time.*
- *The nominated owner must then accept ownership in order for it to be transferred to them.*
- *As the veUNO and Timelock contracts were not included within the scope of this audit, we are unable to provide an assessment with regards to their security or functionality.*

Resolver Contract:

- *This contract is used to calculate pending APY rewards and is not used by the VeUnoDaoYieldDistributor contract.*
- *Rewards are calculated as any APY interest that exceeds the future yield to be earned in the remainder of the VeUnoDaoYieldDistributor's yield period.*
- *The owner can update the APY to up to 100% at any time.*

AUDIT RESULTS

Vulnerability Category	Notes	Result
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	• The owner or Timelock address can pause the contract at any time, preventing all yield claiming. • The owner or Timelock address can update the Timelock address at any time. • The owner or Timelock address can manually update the yield rate to any amount at any time.	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
Improper Events	N/A	PASS
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
Sybil Attack	N/A	PASS
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

ABOUT SOURCEHAT

SourceHat (formerly Solidity Finance - founded in 2020) has quickly grown to have one of the most experienced and well-equipped smart contract auditing teams in the industry. Our team has conducted 1700+ solidity smart contract audits covering all major project types and protocols, securing a total of over $50 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 SOURCEHAT AUDIT?

Typically, a smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. A *SourceHat 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 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