SOLIDITY FINANCE

VeneraSwap - Smart Contract Audit Report

AUDIT SUMMARY

VeneraSwap is building a new decentralized automated liquidity protocol trading platform that is used to exchange tokens, with additional contracts offering the ability to stake for rewards.

For this audit, we reviewed the following contracts on the Binance Smart Chain Mainnet:

- Venera Contract at 0x94174f59c009f49b6aBC362706fdA402616b0427.
- Masterchef Contract at 0xE947062374759D9Dad48B375030099b1ADe1a9c7.
- VeneraRouter Contract at

- VeneraFactory Contract at 0x95F9c44fA1585811e1D1a0F59e74174B657B37A5.
- GaugeProxy Contract at 0x9e0a80cE7dDeD3DB40526e9c961CA5287B55ce81.
- VSWMaker Contract at 0xDd3c85158d209f4E709dF77042Dff99571b7cE10.

AUDIT FINDINGS

> *Please ensure trust in the team prior to investing as they have substantial control in the ecosystem.*
> *Date: January 28th, 2022.*
> *Updated: April 27th, 2022 to reflect new mainnet addresses.*
> *Updated: May 24th, 2022 to reflect new mainnet addresses.*
> *Updated: June 9th, 2022 to reflect new mainnet addresses.*
>
> ### Finding #1 - VSWMaker - Low
>
> ***Description:*** *The swapping done within the contract is open to potential frontrunning.*
> ***Risk/Impact:*** *A frontrunner could order transactions causing a change in price presenting an arbitrage opportunity. This is unlikely on BSC.*
> ***Recommendation:*** *Implement a Time Weighted Average Price Oracle (TWAP) or use Chainlink for pricing to mitigate the potential price manipulations that allow frontrunning.*

Finding #2 - Masterchef - Low (Resolved)

Description: *The Masterchef deposit() function does not properly follow the check-effects design pattern.*

Risk/Impact: *The deposit() function is open to potential reentrancy issues from tokens with fallback functions.*

Recommendation: *Restructure the logic of the deposit() function to call all token transfers at the end of the function or ensure to not utilize ERC-777 or other tokens will fallback functions.*

Resolved: *The team has restructred the logic to avoid reentrancy.*

Finding #3 - Masterchef - Informational

Description: *If a pool's reward token is updated before all tokens have been distributed and no other pools are using that token as rewards, any remaining tokens are stuck in the contract.*

Risk/Impact: *The effective circulating supply of the token will be lowered, and the team will take a loss of the value of the tokens.*

Recommendation: *Exercise caution to ensure that tokens are properly distributed before updating the rewards tokens to prevent tokens being locked.*

Description: *The setPID() function is non-functional after the first use.*

Risk/Impact: *Once a $mxVSW token is deposited into a Masterchef through the GaugeProxy, there is no way to withdraw it. As a result, changing the pool ID will cause the GaugeProxy to be unable to claim rewards from the previous pool and disrupt the reward flow to the Gauges.*

Recommendation: *The setPID() function should only be called once after deployment.*

Finding #5 - Masterchef - Informational

Description: *The owner can set the reward token for any pool to any token at any time.*

Risk/Impact: *If a reward token is added that is also being used as a staking token, users' rewards can be funded with staked tokens.*

Recommendation: *The team should exercise caution to not allow the same token to be used for staking and rewards, as users may withdraw another user's tokens as rewards if insufficient tokens are supplied for rewards.*

Finding #6 - Masterchef - Informational

> ***Risk/Impact:*** *If the staking token is changed, users funds may be locked in the contract and unable to be withdrawn.*
>
> ***Recommendation:*** *The team should exercise caution to not change the staking token for a pool prior to allowing users to withdraw their staked funds.*

CONTRACTS OVERVIEW

> - *The contracts use SafeMath to prevent underflow/overflow issues.*
> - *The project team should exercise caution when assigning staking tokens throughout the platform to avoid fee-on-transfer or ERC777-compliant tokens (this is uncommon).*
>
> *VeneraERC20 Contract:*
>
> - *The VeneraERC20 contract implements the ERC20 standard for use as an LP token.*
> - *This contract utilizes a "permit" mechanism which allows the owner of the $Venera-LP tokens to sign a transaction that enables another user to withdraw tokens and send them to the recipient. The recipient then submits the permit on behalf of the owner.*
>
> *VeneraFactory Contract:*
>
> - *The VeneraFactory contract is responsible for the creation of*

- *When creating a new trading pair, the VeneraPair initialize() function is called which allows the factory to specify the two ERC20 tokens that this pair will exchange.*
- *Once the pool is created, its address is stored with a double mapping that takes both token addresses as input.*
- *The Fee To can disable all swapping on the platform at any time.*

VeneraPair Contract:

- *The VeneraPair contract is the core VeneraSwap trading functionality.*
- *Each VeneraPair manages a liquidity pool made up of reserves of two ERC-20 tokens.*
- *This contract is responsible for tracking the balance of both tokens in the pair, as well as mints and burns of the LP token.*
- *Users can add liquidity by providing an equivalent value of each token and are minted an LP token in return. The LP tokens may be burned to receive the underlying assets at any time.*
- *Users may also exchange one token for an equivalent amount of the other token based on the current market value. A 0.25% fee is taken on an exchange between tokens and given as rewards to LP providers.*
- *Of the 0.25% fee collected, a portion is taken on liquidity adds and burns.*

- *The VeneraRouter contract is used to interact with liquidity pools that are created via the VeneraFactory contract.*
- *VeneraRouter routes orders to the user-determined pair contract to swap assets.*
- *This contract performs requirement checks needed for swapping tokens, adding liquidity, and removing liquidity.*

Venera Contract:

- *$VSW tokens are earned as rewards for interacting with the VeneraSwap platform.*
- *The owner may mint any amount of VSW to any address at any time.*
- *The owner may burn any amount of VSW in the owner's wallet at any time.*
- *The contract is BEP20 compliant and all standard functionality is present.*

MasterDill Contract:

- *The owner is minted one token upon deployment.*
- *There is no burn functionality present, though the circulating supply can be reduced by sending tokens to the 0x..dead address, if desired.*
- *The contract is BEP20 compliant and all standard functionality is present.*

- *Any user may deposit specified tokens into staking pools to earn rewards in the form of another token.*
- *There is a deposit fee, which is transferred to a Fee address controlled by the team. Each pool has its own deposit fee and may vary between pools.*
- *When distribution is enabled, users will receive a reward amount on each block based on the rewards emission rate, the amount staked, and the amount of points allocated to the pool.*
- *On deposits and withdrawals, pending rewards are calculated and transferred to the user. An additional 10% of calculated rewards are transferred to a Dev address controlled by the team.*
- *If minting is enabled by the owner, the reward tokens are minted to the contract until claimed by the user. An additional 10% of the rewards are minted to the Dev address.*
- *If a referrer was provided when deposited and the referrer has 100 $VSW tokens, 5% of the calculated rewards are transferred to the referrer.*
- *Additionally, if the previous "level 1" referrer was also referred and that "level 2" referrer has 1000 $VSW tokens, the level 2 referrer will receive an additional 2% of the calculated rewards.*
- *If there is no level 1 or level 2 referrer, or they do not have sufficient tokens, the 5% and 2% respectively will be transferred to the Dev address.*

their address, without calculating rewards.

- *The Dev address may withdraw any $VSW tokens in the contract at any time.*
- *The owner may add a new pool at any time.*
- *The owner may update the rewards emission rate, points allocated to each pool, and deposit fee to any value at any time.*
- *The owner may enable and disable reward distribution at any time.*
- *The owner may enable and disable emergency withdraws for any pool at any time.*
- *The owner may update any pool's reward token and staking token at any time.*
- *The team should exercise caution to not allow the same token to be used for staking and rewards, as users may withdraw another user's tokens as rewards if insufficient tokens are supplied for rewards.*
- *The team must also ensure not to add the same token twice for staking in order to properly calculate reward per share.*
- *The Dev and Fee addresses may update their own address at any time.*
- *If the Masterchef is the owner of the VSW token contract, they may transfer ownership to any address at any time.*

Gauge Contract:

- *Any user may deposit a specified token in order to earn*

- *A deposit fee is taken and transferred to a Treasury address controlled by the team.*
- *During the rewards period, users will receive a reward amount on each block based on the rewards emission rate and the amount staked.*
- *Rewards are not distributed when depositing or withdrawing and must be separately claimed.*
- *Users may elect to "exit" the contract, withdrawing their entire balance and claiming rewards.*
- *Rewards must be supplied from an external "distributor" contract. The rewards period extends 7 days from each rewards distribution.*

GaugeProxy Contract:

- *This contract serves as a valid distributor for any Gauge contract and also keeps track of users' voting power accross multiple tokens.*
- *A Governance address used to manage the contract and a Treasury address where Gauges send fees are declared upon deployment.*
- *A user's votes for each token are calculated as a percentage of the user's total token balance across all tokens.*
- *Users may "reset" their votes, setting their votes to 0 across all tokens. They must then make another call to vote in order to recalculate their votes per token.*
- *Rewards are earned by depositing $mxVSW into a Masterchef*

- *The Governance address may add a Gauge for any token that does not already have a Gauge, as long as a valid Treasury address has been set.*
- *The Governance address may set the Masterchef pool ID that the contract deposits into at any time.*
- *The Governance address may update the deposit fee rate that is used for all Gauges at any time, up to a maximum of 20%.*
- *The Governance address may update the Treasury address at any time.*

VSWMaker Contract:

- *Any user may transfer LP tokens to this contract in order to convert them to $VSW, which is subsequently sent to a destination address.*
- *Once LP tokens have been sent to the contract, any externally owned address (EOA) may trigger the burning of the LP tokens.*
- *The contract will then burn the LP tokens for the underlying assets, swapping any non-VSW token for $VSW using a bridge.*
- *A fee is taken from the final $VSW amount and sent to a Fee address controlled by the team. The remaining $VSW is subsequently sent to a destination address, which is set upon deployment.*
- *Users also have the option to burn multiple different specified LP tokens at once.*
- *The owner may set and update the bridge for any token at any*

> - *The owner may withdraw any token in the contract at any time.*
> - *The owner may update the fee percent and the Fee address at any time.*

EXTERNAL THREAT RESULTS

Vulnerability Category	Notes	Result
Arbitrary Storage Write	N/A	PASS
Arbitrary Jump	N/A	PASS
Centralization of Control	• The team can disable all trades on the VeneraSwap DEX at any time. • The team retains ownership functionality described above. • There are multiple instances of uncapped fees.	WARNING
Delegate Call to Untrusted	N/A	PASS

Vulnerability Category	Notes	Result
Dependence on Predictable Variables	N/A	PASS
Deprecated Opcodes	N/A	PASS
Ether Thief	N/A	PASS
Exceptions	N/A	PASS
External Calls	N/A	PASS
Integer Over/Underflow	N/A	PASS
Logical Issues	N/A	PASS
Multiple Sends	N/A	PASS
Suicide	N/A	PASS
State Change External Calls	N/A	PASS

Vulnerability Category	Notes	Result
Unchecked Retval	N/A	PASS
User Supplied Assertion	N/A	PASS
Critical Solidity Compiler	N/A	PASS
Overall Contract Safety		PASS

CONTRACT SOURCE SUMMARY AND VISUALIZATIONS

Name	Address/Source Code	Visualized (Hover-Zoom Recommended)
GaugeProxy	BSC Mainnet	Function Graph.

VeneraFactory	BSC Mainnet	Function Graph. Inheritance Chart.
VeneraRouter	BSC Mainnet	Function Graph. Inheritance Chart.
Venera	BSC Mainnet	Function Graph. Inheritance Chart.
VSWMaker	BSC Mainnet	Function Graph. Inheritance Chart.
MasterChef	BSC Mainnet	Function Graph. Inheritance Chart.

GO HOME