

STATE MIND

veYFI (Voting and Reward pool)

04-07-2022 - 08-07-2022



1. Project Brief		3
2. Finding Severity breakdown		4
3. Summary of findings		5
4. Conclusion		5
5. Findings report		6
Critical	veYFI balance might not decrease	6
	Incorrect balance calculation leading to higher reward	6
High	Incorrect interface	6
	Incorrect balance calculation	7
Medium	Tokens may be lost if checkpoint functions are not called for 20 weeks	7
	Incorrect indentation	7

Informational	Function <code>_find_timestamp_epoch()</code> can be view	7
	Code duplication	8
	Excess reentrancy protection	8
	No check for address zero	8
	Unnecessary condition	8
	Unreachable code	9
	Unnecessary gas usage	9
	Unnecessary check	9
	Incorrect comment	9
	Unused arguments	9

1. Project Brief



Title	Description
Client	Yearn
Project name	veYFI (Voting and Reward pool)
Timeline	04-07-2022 - 08-07-2022
Number of auditors	3
Initial commit	696bb76be86a601f25cda577bfb9dc14daa91079
Final commit	bb9d8ac9dd90a9a9772b9663ce4fa232fda7bce2


Short Overview


veYFI is the token locking mechanic similar to the ve-style program of Curve. YFI tokens can be locked for any time, but the max reward generates when locking time is more or equal than 4 years. There are several contracts in the auditing scope:

- **VotingYFI** contract is intended to calculate voting power at any time point based on the locked YFI amount.
- **RewardPool** contract performs the distribution of YFI rewards based on voting power. The time axis is divided into weeks, i.e. any changes apply once per week (e.g. locked amount changes, top-up tokens for rewards).

Project Scope

The audit covered the following files:

 [VotingYFI.vy](#)

 [RewardPool.vy](#)

2. Finding Severity breakdown



All vulnerabilities discovered during the audit are classified based on its potential severity and has the following classification:

Severity	Description
Critical	Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party.
High	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.
Medium	Bugs that can break the intended contract logic or expose it to DoS attacks, but do not cause direct loss funds.
Informational	Bugs that do not have a significant immediate impact and could be easily fixed.

Based on the feedback received from the Customer regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect its security.
Acknowledged	The Customer is aware of the finding. Recommendations for the finding are planned to be resolved in the future.

3. Summary of findings

Severity	# of Findings
Critical	2
High	2
Medium	2
Informational	10

4. Conclusion

2 critical, 2 high, 2 medium and 10 informational severity issues were found, fixed all critical, high, medium and 4 informational issues were acknowledged.

Deployment commit: [bb9d8ac9dd90a9a9772b9663ce4fa232fda7bce2](#)

Deployment

File name	Contract deployed on mainnet
contracts/VotingYFI.vy	Ox9Oc1f922Od9Od3966fbee24O45edd73e1d588ad5
contracts/RewardPool.vy	Oxb287a1964aee422911c7b84O9f5e5a273c1412fa

5. Findings report



Critical

veYFI balance might not decrease	Fixed at PR#179
<p>Description</p> <p>It is possible for a user to create a lock longer than <code>MAX_LOCK_DURATION</code>, but such that <code>kink.ts</code> at the line: VotingYFI.vy#L136 is less than <code>block.timestamp</code> since it rounds down to a week. Then in the function <code>balanceOf()</code>, function <code>replay_slope_changes()</code> will not consider the change of slope at <code>kink.ts</code> resulting in a balance of user that doesn't decrease with time.</p> <p>Recommendation</p> <p>It is recommended to refactor how timestamps are rounded to weeks and check if all slope changes are applied correctly.</p>	
Incorrect balance calculation leading to higher reward	Fixed at PR#182
<p>Description</p> <p>At the line: RewardPool.vy#L266</p> <p>The balance calculation can be incorrect if a user creates a lock longer than 4 years such that checkpoint slope is <code>0</code>. Then <code>balance_of</code> would be the initial veYFI balance at <code>old_user_point.ts</code> without factoring change of slope in <code>kink</code>. Then a user might get a bigger reward than they were supposed to.</p> <p>Recommendation</p> <p>It is recommended to use function <code>balanceOf</code> from the <code>VotingYFI</code> contract.</p>	

High

Incorrect interface	Fixed at 4e9dfe
<p>Description</p> <p>At the lines: RewardPool.vy#L9-L16</p> <p>The interface for the <code>VotingYFI</code> contract is incorrect.</p> <p>Recommendation</p> <p>It is recommended to update the interface and the function calls to that contract.</p>	

Incorrect balance calculation	Fixed at PR#182
<p>Description</p> <p>At the line: RewardPool.vy#L175</p> <p>The function <code>ve_for_at()</code> returns the veYFI balance for a user at the timestamp. But the calculation can be incorrect if a user creates a lock longer than 4 years such that the checkpoint slope is <code>0</code>. Then the balance at <code>_timestamp</code> would return the initial veYFI balance without factoring change of slope in kink.</p> <p>Recommendation</p> <p>It is recommended to use function <code>balanceOf</code> from the <code>VotingYFI</code> contract.</p>	

Medium

Tokens may be lost if checkpoint functions are not called for 20 weeks	Fixed at cfd19
<p>Description</p> <p>If the function <code>_checkpoint_token()</code> is not called for more than 20 weeks, then <code>since_last</code> at the line RewardPool.vy#L92 will be bigger than 20 weeks, but the <code>for</code> loops only adds rewards for 20 weeks, meaning some tokens will be lost.</p> <p>If the function <code>_checkpoint_total_supply()</code> is not called for more than 20 weeks, then when claiming rewards <code>self.ve_supply[week_cursor]</code> will reach <code>0</code> reverting the transaction.</p> <p>Recommendation</p> <p>It is recommended to refactor how rewards are distributed in edge cases and increase weeks in <code>_checkpoint_total_supply()</code>.</p>	

Incorrect indentation	Fixed at 961ef6
<p>Description</p> <p>RewardPool.vy#L211 With incorrect indentation the contract won't compile.</p> <p>Recommendation</p> <p>It is recommended to correct indentation.</p>	

Informational

Function <code>_find_timestamp_epoch()</code> can be view	Fixed at PR#190
<p>Description</p> <p>At the line RewardPool.vy#L130 function can be view.</p> <p>Recommendation</p> <p>We recommend adding <code>@view</code> modifier for this function.</p>	

Code duplication	Fixed at PR#182
------------------	---------------------------------

Description

Function `_find_timestamp_epoch()` at [RewardPool.vy#L130-L142](#) is similar to `_find_timestamp_user_epoch()`. Call `_find_timestamp_epoch(timestamp)` is equal to `_find_timestamp_user_epoch(VEYFI.address, timestamp, VEYFI.epoch(VEYFI.address))`.

Recommendation

We recommend rewriting `_find_timestamp_epoch()` function:

```
@view
@internal
def _find_timestamp_epoch(_timestamp: uint256) -> uint256:
    return _find_timestamp_user_epoch(VEYFI.address, _timestamp, VEYFI.epoch(VEYFI.address))
```

Excess reentrancy protection	Acknowledged
------------------------------	--------------

Description

At the line [RewardPool.vy#L284](#) `@noreentrant('lock')` is not needed because this function only calls YFI token contract and VotingYFI contract.

Recommendation

We recommend removing `@noreentrant('lock')`.

Client's comments

no harm

No check for address zero	Acknowledged
---------------------------	--------------

Description

At the lines: [VotingYFI.vy#L87-L88](#)
There is no check for address zero for parameters `token`, `reward_pool`.

Recommendation

It is recommended to add sanity checks.

Client's comments

yearn thinks this issue can be mitigated during proper ops deployment of contract risk and impact are low since if addresses are not configured contract can be redeployed

Unnecessary condition	Acknowledged
-----------------------	--------------

Description

At the line: [RewardPool.vy#L100](#)
The check `block.timestamp == t` is unnecessary since if `since_last == 0`, then `t` is equal to `block.timestamp`.

Recommendation

It is recommended to remove this condition.

Unreachable code	Acknowledged
<p>Description</p> <p>At the line: RewardPool.vy#L106</p> <p>The condition is never true since <code>next_week</code> can never be equal to <code>t</code>. If <code>since_last == 0</code>, then <code>block.timestamp < next_week</code>.</p> <p>Recommendation</p> <p>It is recommended to remove the if condition.</p>	
Unnecessary gas usage	Fixed at PR#193
<p>Description</p> <p>At the line: RewardPool.vy#L86</p> <p>In the function <code>_checkpoint_token()</code>, if <code>to_distribute</code> is <code>0</code>, then the <code>for</code> loop at line RewardPool.vy#L97 would not change anything.</p> <p>Recommendation</p> <p>It is recommended to check if <code>to_distribute</code> is <code>0</code> before the loop.</p>	
Unnecessary check	Fixed at fc3536
<p>Description</p> <p>RewardPool.vy#L212 The uint256 <code>balance_of</code> var is already compared to zero, so it isn't necessary to check if <code>balance_of > 0</code>.</p> <p>Recommendation</p> <p>It is recommended to remove the check.</p>	
Incorrect comment	Fixed at PR#189
<p>Description</p> <p>There is no contract owner, so anyone can call <code>checkpoint_token</code> anytime at RewardPool.vy#L121-L122</p> <p>Recommendation</p> <p>We recommend deletion of a part the comment about the owner</p>	
Unused arguments	Fixed at 4e9dfe
<p>Description</p> <p><code>ve</code> isn't used at RewardPool.vy#L130 RewardPool.vy#L147</p> <p>Recommendation</p> <p>We recommend the deletion of unused <code>ve</code></p>	

STATE
MIND