G0 Group

Security Review of

Nibbl

February 2022

Nibbl / February 2022

Files in scope

The following solidity files in:
https://github.com/NibblNFT/smart_contracts/tree/bc68df00a4e227c404eed5faa5736eef92969504/contracts

- NibblVaultFactory.sol
- NibblVault.sol
- Basket.sol
- Utilities/NFT.sol
- Twav/Twav.sol
- Proxy/ProxyVault.sol
- Bancor/BancorBondingCurve.sol
- Bancor/Power.sol

Current status

All found issues have been fixed or addressed.

Issues

1. Curve fee can push reserve ratio in secondary curve over 100%

Severity: major

If transaction on primary curve is large enough and secondary curve reserve ratio is < 50%, it's possible to increase secondary RR > 100%, which will break selling and buying along secondary curve.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

2. Admin can re-enter contract when receiving fee breaking accounting

Severity: major

In a buy call, admin can re-enter the contract with a sell call when receiving fee. This allows for example to reduce supply below the threshold for secondary curve, with primary curve buy being finished after. This will break contract's accounting.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

3. Bidder can buy shares for free

Severity: critical

Funds used to buy shares above the bid valuation go to the bidder if the bid goes through, this allows bidder to essentially buy shares for free allowing him to steal from other shareholders. Bidder can also place the bid above the current valuation, buy shares up to the rejection veluation and then sell them off to counter price increasing buys from other users, manipulating the rejection process while making a profit.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

4. Curator can redeem fee multiple times due to re-entrancy

Severity: critical

`redeemCuratorFee` is open to a reentrancy attack, because `feeAccruedCurator` is zeroed after the external call. This allows the curator to redeem their fee multiple times.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

5. Admin can DoS buying and selling

Severity: major

Because fee is transferred to admin address during `sell` and `buy` calls and the calls are required to succeed, the contract at the admin address can block the execution of the `NibblVault` contract.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

6. Discontinuity in buyout periods

Severity: minor

When `block.timestamp == buyoutEndTime` the nft is neither `boughtOut` nor `notBoughtOut`.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

Additional issues

These issues were discovered in the second round of auditing after some changes to the original code were introduced. The audited code is in this commit:
https://github.com/NibblNFT/smart_contracts/tree/bc68df00a4e227c404eed5faa5736eef92969504/contracts

7. Anybody can reset NibblVaultFactory configuration

Severity: critical

There's an issue with how config vars are updated in `NibblVaultFactory` for example `feeToUpdateTime` starts as `0` so it's `< block.timestamp` by default, anybody can call `updateNewAdminFeeAddress` from the start and reset it to `address(0)`, same goes for all other config vars.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

8. Excess curve fee not retuned to the buyer/seller

Severity: medium

In `_chargeFee` when curve fee would lead to reserve ration rising above 50%, the curve fee is lowered by the necessary amount to prevent this, the `_feeCurve` variable itself however isn't lowered, this is an issue because this variable is used to calculate the amount that is left for buy/sell transaction after fees. This means that the amount will end up being artificially lower than it should be.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

9. lastBlockTimeStamp needs to be reset when twavObservations array is deleted

Severity: medium

After every unsuccesful bid, `twavObservations` array is deleted, effectively replacing all observations with valuation `0` at timestamp `0`, when first twav observation is added after that, it's calculated from `lastBlockTimestamp` which still contains timestamp of the last, now deleted, observation from previous bid. This mismatch between observations being set to timestamp `0` and `lastBlockTimestamp` being much higher will lead to atrificially lower `_getTwav()` return value until all observations are replaced.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts

Additional issues #2

These issues were discovered in the third round of auditing after additional changes to the original code were introduced. The audited code is in this commit:
https://github.com/NibblNFT/smart_contracts/tree/9d8136d73e231e19fdffe72e6cea584c06da4719/contracts

10. Nonce in permit function not incremented

Severity: critical

`NibblVault.permit` function doesn't increment the `_nonces[owner]` counter which allows permit transactions to be replayed.

status - fixed

The issue is no longer present in:
https://github.com/NibblNFT/smart_contracts/tree/8589bfcbc28e352ec3e55577262f6cd17c0ec293/contracts