Decurity Checklists

Summary

Author: Decurity

Source: audit-checklists

Automated Market Maker protocols (AMM) Checklist

General

- □ Does the AMM use any forked code, e.g. from Uniswap code base? Use contract-diff.xyz to compare the code.
- □ Are there any potential problems with rounding in formulas of product constants?
- □ `transfer()`, `transferFrom()` and even `safeTransfer()` functions can lead to the re-entrancy if called on untrusted tokens or hook-on-transfer tokens such as ERC777 and ERC677. And even if the `nonReentrant` modifier is used, a cross-contract view re-entrancy attack is possible. Is CEI pattern followed when updating `reserves[]`?
- □ Most AMMs support flashloan functionality (e.g. FlashSwaps in Uniswap), is callback function called after token transfer and not before?
- □ In some tokens `transfer()` and `transferFrom()` may return `false` instead of `revert()`. Is the return value of these functions checked?
- □ Does the AMM update the user's token balances using signed integer? Can `-int(amount)` lead to an integer overflow?
- □ Does the AMM support fee-on-transfer tokens? Is this fee taken into account in the swaps?
- □ Can arbitrary calls be made from the user input?

TWAMM

- □ Does the TWAMM support rebasing tokens? What happens if during a long-term swap there is a balance change?
- □ Is there a check for the sufficient liquidity before the swap?

Integration

- □ Callback functions in the external contracts that are called by the `swap()` function are required to check the address of the calling contract. Does the contract that integrates an AMM callback have this check? Are there any ways to bypass the check?
- □ Protocols that integrate AMMs for token swaps should calculate the `minAmountOut` before the swap. Are external oracles used? Is a sandwich attack possible?

Collateralized Debt Position (CDP) Checklist

General

- □ If a user's CDP is closed (e.g., when the user has paid off all the debt), what happens to the vault entry in storage? Is the storage variable erased? Are there checks in the code sections where the variable is used to ensure its existence?
- □ Is it possible to have a condition when a user cannot repay their loan?
- □ How is the pool value calculated? How is the fee paid by borrowers for the loan distributed? If a portion goes to the lender (depositor) and another portion goes to the pool, is the calculation of the pool value correctly implemented?
- □ If ERC4626 is implemented in the protocol, is there a possibility of vulnerability to an inflation (first deposit) attack?

Collaterals

General

- □ Some tokens have a black list (e.g. USDT). How can this affect the protocol?
- □ Some tokens can be `pause()`d. How can this affect the protocol?
- □ Does the protocol consider the implementation differences of the `transfer()` function for tokens, such as BNB's restriction on transferring to a zero address or transferring an amount of zero?
- □ The transferred token can take a fee for the transfer. How can this affect the protocol?
- □ Some tokens may not revert in `transfer()` and `transferFrom()`, does the protocol use SafeERC20?
- □ Does the protocol account for the change in account balance caused by rebasing tokens such as aTokens, UBI, AMPL, and similar tokens?

ERC20 tokens

- □ Does the contract consider that tokens in the market can have different decimals? How does this affect the calculation of the collateral value?
- □ If ERC777, ERC721, ERC677 or other ERCs with callbacks are used in the market, is a reentrancy attack possible?
- □ Is it possible to deposit one stablecoin as collateral in the protocol and withdraw a different stablecoin? Is there a potential for arbitrage in this scenario?
- □ How does the protocol calculate the value of the same stablecoin? Does the price calculation take into consideration the possibility of a depeg or deviation from the intended pegged value?

NFT: liquid ERC721 tokens from the BAYC, MAYC, CryptoPunks, Azuki and other collections

- □ If a CryptoPunks collection that does not support the `transferFrom()` function is used as collateral, is it safe to transfer tokens to the vault? When calling the `offerPunkForSaleToAddress()` function, there is a possibility that ownership of the token can be front-runned.
- □ If `onERC721Received()` is implemented in ERC721, is re-entrancy possible?

Uniswap LP positions

- □ If the protocol allows the use of LP tokens as collateral, then how does it determine the value of these LP tokens? Red flag-`pool.getReserves()`

- □ What type of LP token(0.01%, 0.05%, 0.3%, 1%) is used in the protocol? Do the contracts consider that there are several pairs for the same tokens in Uniswap?

Earn tokens

- □ If tokens that are pegged to any asset (renBTC, HitBTC, aBTC, etc.) are used in the market (for example, as collateral), then how will the protocol behave during the depeg? Will it count the price 1:1?
- □ When the protocol integrates earn protocols for earning on collateral, is the mechanics of calculating shares implemented correctly?
- □ How is the user's share counted if the collateral is staked somewhere? Is it possible to manipulate the calculation?

Oracles

General

- □ Can the asking price be zero? Is there any processing of such a case?
- □ Are the decimals of the received prices processed correctly?

Chainlink

- □ Is getting prices wrapped in a try/catch? Are there alternative ways to get a price, considering that multi-signature wallets in Chainlink can block access to data feed?

Uniswap TWAP

- □ TWAP is subject to the manipulations, especially if a small time window is used. Is TWAP manipulation possible? How much will it cost to the attacker?

Virtual Price (amountTokenA/amountTokenB)

- □ How do external non-protocol oracle contracts calculate the price? Special attention should be paid to the price of vToken/Token. Red-Flag-`balanceOf()` in the price calculation method trace.

Markets

Hybrid markets

- □ If one pool is hacked, how will it affect other pools? Does the protocol isolate liquidity?

Interest rate

General

- □ When is the interest rate calculated? Before or after closing the vault?

Liquidations

General

- □ Can the user liquidate their own position with a profit? How can the health ratio change, apart from typical cases such as withdrawal of collateral, taking on additional debt, or a significant drop in the price of collateral?
- □ When is the interest rate calculated? Before or after liquidation of the vault?
- □ Is there a risk of re-entrancy in the ERC721 withdrawal method before checking for a normal health ratio? If the health ratio is checked after calling safeTransferFrom() with the implementation of onERC721Received(), is there any way to bypass the health ratio check in the callback?

Auction liquidations

- □ If an auction is used during liquidation and the liquidated person has the ability to prematurely complete the auction by proving their solvency, can he or she take a flashloan in the process?
- □ Is there enough input data validation in the auction start function? Is it possible to launch an unfinished auction?
- □ Is the math correct when only a portion of the collateral from the liquidated vault is put up for auction?
- □ If users send tokens to a contract when creating a bid in an auction, what happens if their bid is interrupted? Is the interrupted person's money returned to them? What happens if the creator of the auction prematurely completes it (for example, by repaying the entire debt)? Is the bid returned to the last bidder?

Liquid Staking Derivatives (LSD) Checklist

General

- □ How is data inserted into the validators registry? How are `WithdrawCredentials` set for the validator in **DepositContract**? Is it possible that a deposit transaction is frontrunned by a malicious validator to set immutable `WithdrawCredentials`?
- □ The derivatives must be collateralized by 100% ETH. However, the amount of ETH may decrease due to slashing. Does the implementation of the derivative token support burning? Is a depeg between the *Derivative* and *UnderlyingToken* possible?
- □ TVL (Total Value Locked) of LSD protocols is the largest among all types of DeFi systems. Centralization of key functions leads to great risk. An ideal LSD protocol should be fully decentralized (e.g., support multisig).Who can add validator data to the registry? Who can mint or burn derivatives? Who can add roles for addresses? Is it a single address or a multisig?
- □ How often is `DepositContract.deposit()` called? Who initiates the function call? If a significant amount of ETH has accumulated in the protocol contract balance and `DepositContract.deposit()` is called, how many iterations of the call can be executed? How much gas will these iterations require? At what amount of ETH will an out-of-gas issue occur on the contract?
- □ Are there operations in contracts where an array with data from all validators is iterated? How much gas does each iteration require? Is an out-of-gas issue possible with a large number of array elements (i.e., a large number of validators)? Example
- □ Is there a possibility of exploiting an inflation (first deposit) attack? How are empty pools created? When creating an empty pool, are underlying tokens deposited? Is it possible to manipulate the price

of shares immediately after creating a pool? A good solution to avoid an inflationary attack is to create a user with a non-zero balance when creating a pool.

- ☐ Is slashing of the operator's balance possible in the protocol? What happens if the amount of the slashing penalty is greater than the operator's balance? Can the operator withdraw the balance of collateral tokens while continuing to validate blocks?
- ☐ Special attention should be paid to the functions of withdrawing staked ETH. Is a reentrancy attack possible during withdrawal?
- ☐ Is a reentrancy attack possible in the user's rewards distribution function?
- ☐ Does the accounting of user rewards work correctly? If the protocol calculates the user's reward based on their share of the pool, are there any issues with the calculation of the share?
- ☐ Who determines the price of the derivative token? If the price is determined by oracles, is it possible to manipulate the price via the sandwich attack when a price update transaction is sent?