Ideamarket

Quantstamp Audit Response

31st January 2021

QSP-1: Possible Incorrect Uniswap Pair

Fixed. 3-hop paths have been added to `MultiAction`.

QSP-2: Unclaimed COMP Tokens

Fixed. `InterestManagerCompound.withdrawComp` now calls `claimComp` on Compound's `Comptroller`.

QSP-3: Unchecked function arguments

Fixed except for `priceRise` in `IdeaTokenFactory.addMarket.` Ideamarket might in the future add markets with a constant price (`priceRise = 0`). The missing check in this case is intended.

QSP-4: Inconsistent use of `accrueInterest`

Fixed. `IdeaTokenExchange.withdrawTradingFee` now calls `accrueInterest.`

QSP-5: Privileged Roles and Ownership

The contract system has been designed with upgradeability in mind. All points listed are intentionally designed the way they currently are:

1. *The owner of* `IdeaToken` *can mint arbitrarily*: The owner of `IdeaToken` is `IdeaTokenExchange` which mints and burns when tokens are sold and bought. Arbitrary access to `mint` would need a contract code change which is protected by the `Timelock.`

2. *The owner of* `ProxyAdmin` *can change the contract logic arbitrarily*: Having the option to upgrade contract code allows Ideamarket to effectively add new features in the future. Contract code changes are protected by the `Timelock.`

3. *The owner of* `IdeaTokenExchange` *can invoke* `setTokenOwner` *and* `setPlatformOwner` *arbitrarily*: Both functions are used to store the address of the owner of a listing (for example a Twitter account) or platform (for example Twitter) on-chain. As this data is not available on-chain, Ideamarket runs a verification service which handles first-time verification of accounts in the system (via the `authorizer` address).
   After an owner has initially been set, the authorizer is not allowed to change the owner address anymore. Ideamarket still has the possibility to change the owner of the listing, however now the `Timelock` applies (see below). This has been built in as Ideamarket will also target crypto newcomers, like publishers which want to use Ideamarket to create an income stream but are inexperienced with using Ethereum. We expect these newcomers to often lose access to their private keys. In this case Ideamarket can, after thorough verification, restore access to the listing.

4. *The owner of* `IdeaTokenFactory` *can set the trading and platform fees arbitrarily, and also change name verifiers*: The trading fees are Ideamarket's main income and thus need a way to be dynamically adjusted. Ideamarket might for example decide to decrease the trading fee for a certain market and increase the platform fee by the same amount, thus directing more fees towards the listed platform.
   Name verifiers can be changed in case a name verifier is not acting correctly. For example, a name verifier might disallow a valid name due to a bug in its code. In this case the name verifier can be updated with a correct implementation
   Again the `Timelock` applies, see explanation below.

5. *The owner of* `InterestManagerCompound` *may invoke the* `redeem*` *functions at any point*: The owner of `InterestManagerCompound` is `IdeaTokenExchange` which invests/redeems Dai when `IdeaTokens` are bought or sold. Arbitrary access to the above function would need a contract code change which is protected by the `Timelock.`

All changes made to the system, including contract code changes, need to go through the `Timelock (DSPause)` which assures that upcoming changes are publicly visible as queued on-chain for a certain time until they can be executed. Additionally, the access to the `Timelock` is protected by a 2-of-2 *Gnosis Safe Multisig* controlled by the Ideamarket team.

QSP-6: Unchecked external function call

Fixed. The call has been wrapped in a `require` statement.

QSP-7: Incorrect while-loop condition

Fixed. The invalid condition has been removed.

QSP-8: Clone-and-own

We have deliberately chosen the clone-and-own pattern for the listed contracts. While we agree that this method is not always the best one to use and that it has certain downsides, we believe that in our case it is preferable:

- Some of the contracts have been modified or extended. This would not have been possible by inheriting from the original upstream contracts as some of them have private state variables and functions which we needed to modify.

- Cloning removes the risk of unknowingly pulling upstream changes into the codebase. This is especially important when working with the *Transparent Proxy* pattern, where even a simple reordering of state variables can break existing on-chain contracts after an upgrade.

QSP-9: Unlocked Pragma

Fixed. All contracts now use compiler version `0.6.9`.

QSP-10: Allowance Double-Spend Exploit

Acknowledged. We will keep an eye on upcoming community agreements regarding token allowances.

QSP-11: Dependence on external DeFi protocols

We are aware of the dependence on external DeFi protocols which are mainly Compound for interest generation and MakerDAO for Dai. We will follow Quantstamp’s recommendation to improve the user documentation.

QSP-12: Incorrect Token Transfer Logic

Fixed. The function is only intended to be called by the `IdeaTokenExchange`. Since no damage could be done by an external user calling this function any protection has been omitted to save gas. For clarity `onlyOwner` has now been added.

QSP-13: Unclear incentive for vault usage

Currently there is no *direct* financial incentive for vault usage, which is why it is a fully optional feature. From the Ideamarket documentation:

Locking tokens shows your confidence in the long-term value of a listing. It shows future buyers you won't sell as soon as they buy in after you, reducing risk and making the listing more attractive.

The locking feature is intended for long-term investors who want to indicate their confidence about the future success of a listing. By locking tokens, which increases the locked percentage publicly displayed on the Ideamarket frontend, investors can present their trust in the listing which in turn might make it more attractive for subsequent investors.

With that being said, we are currently re-evaluating the vault and might initially launch without it.

QSP-14: Incorrect Domain Name Validation

Rejecting capital letters is intended. Since domain names are case-insensitive we decided on using the lower-case versions. Otherwise, users could list the same domain multiple times: `example.com` and `eXaMpLe.com` which we want to avoid and thus only accept lower-case domain names.