Fuul Protocol - Security Review

Table of contest

Introduction

Fuul team requested the review of the following contracts under the repository **protocol-contracts** referenced by the commit 999fda2e7e7376511c2380b3e2b3ff1fdd772da2: *IFuulFactory.sol*, *IFuulManager.sol*, *IFuulProject.sol*, *FuulFactory.sol*, *FuulManager.sol*, and *FuulProject.sol*.

The contracts under the *rewards_mock* directory are out of the scope of this security review.

The rest of the imported contracts are assumed to be audited.

General Notes

- Gas: Consider using `calldata` instead of `memory` on external functions to reduce gas consumption whenever possible.
- Follow CEI pattern: When the consumption of the gas is still optimal, it is recommended to follow the check effect interaction pattern like in lines 254, 423,

and 481 of *FuulProject.sol* . It is repeated in the rest of the contracts.

Interfaces

IFuulFactory.sol

Notes

- N1 - Parameter names start with `_` for some functions such as `createFuulProject` but without it for others. Consider removing the `_` or adding it to other parameter names.

IFuulManager.sol

Notes

- N1 - Parameter names start with `_` for some functions such as `setClaimCooldown` but without it for others. Consider removing the `_` or adding it to other parameter names.
- N2 - line 3 - Circular dependency with IFuulProject.sol

IFuulProject.sol

Notes

- N1 - Parameter names start with `_` for some functions such as `setProjectURI` but without it for others. Consider removing the `_` or adding it to other parameter names.
- N2 - line 2 - Circular dependency with IFuulManager

Implementations

FuulFactory.sol

Low

- L1 - line 53 - `userProject` is not being used. Consider removing it and its method: `getUserProjectByIndex` and `getUserProjectCount`.

- L2 - lines 375 and 393 - Prevent undesired behaviors by checking if `_period` is greater than 0. If `projectBudgetCooldown` is set to 0, the project admin can remove the budget at any time.

Notes

- N1 - Parameter names start with `_` for some functions such as the constructor but without it for others. Consider removing the `_` or adding it to other parameter names.
- N2 - line 65 - Missing check of input parameters. E.g: `_fuulManager` and `protocolFeeCollector` can be the zero address, `_nftFeeCurrency` and `acceptedERC20CurrencyToken` can't be a contract. Consider adding the missing checks.
- N3 - lines 219, 237, 255, 273, 280, 291, 311, 331, 353, 375 and 393 - Consider emitting events to track changes to the protocol offchain.
- N4 - Typo at `setNftFixedFeeAmounte`. It should be `setNftFixedFeeAmount`

FuulManager.sol

High

- H1 - line 105 - `claimLimitPerCooldown` can be updated and be lower than the `cumulativeClaimPerCooldown`. Consider throwing an error if that happens.
- H2 - line 265 - `_addCurrencyLimit` can allow the admin to accidentally reset the period. `cumulativeClaimPerCooldown` can be higher than 0 at that moment. Consider checking if the `cumulativeClaimPerCooldown` is 0 or the token was already added.

Low

- L1 - line 74 - `claimCooldown` can be set to 0 and therefore bypass `cumulativeClaimPerCooldown` checks. Consider using a minimum to prevent undesired behaviors.
- L2 - lines 89 and 265 - The contract admin can set a currency limit for tokens not accepted in the protocol. Consider checking whether the token is accepted or not.
- L3 - line 111 - The method accepts tokens whose limit has not been set using `addCurrenyLimit` or `_addCurrentLimit`. Consider checking if the token was already set.

- L4 - line 111 - The new limit for a current can't be 0, but `_addCurrencyLimit`, which is being called in the constructor, accepts 0. This makes it impossible to prevent claiming that token if it is a valid scenario. Consider allowing or preventing 0 on both methods.

Notes

- N1 - Parameter names start with `_` for some functions such as the constructor but without it for others. Consider removing the `_` or adding it to other parameter names.
- N2 - line 40 - Missing check of input parameters. E.g: `_attributor` and `_pauser` can be the zero address, `acceptedERC20CurrencyToken` can't be a contract. Consider adding the missing checks.
- N3 - lines 67, 105 and 265 - Consider emitting events to track changes to the protocol offchain.

FuulProject.sol

High

- H1 - line 132 - The Contract can be initialized multiple times. Consider checking if the contract has been initialized or not.
- H2 - lines 225 and 505 - Deposit of special ERC20 tokens will generate an inconsistency in the protocol. For example, when using ERC20 deflationary tokens, the actual amount transferred to the contract will be lower than the `amount` expected. If chances are that the protocol accepts these kinds of tokens, consider adding a pre and post-check of the contract balance to compute the correct amount transferred.
- H3 - lines 643 - Double-attribution-spending. The attributor can use the same proof for different projects. Consider computing the proof in the transaction with individual project parameters. One alternative can be to incude as part of the proof the address of the project contract.
- H4 - line 739 - Possible lock of funds. If the `availableToClaim` value is higher than the `claimLimitPerCooldown`. The user won't be able to claim it. Consider allowing users to claim a limit by parameter.

Medium

- M1 - line 628 - Bypass accepted currencies by attributors - If a specific token has been disabled, the attributor can still attribute transactions for that token. Consider checking if the currency is still accepted when attributing a transaction.

Low

- L1 - lines 209 and 481 - Prevent receiving ETH when the currency is not the 0 address (ETH).
- L2 - lines 254, 433 and 755 - An ERC1155 can support also the ERC721 interface making it impossible to be deposited as ERC1155. E.g: TheSandbox assets.
- L3 - lines 268 and 423 - Bypass `_nonZeroAmount` check For ERC1155 tokens. Consider checking the amount after the sum of the `amounts` array if it is considered important.

Notes

- N1 - Parameter names start with `_` for some functions such as the constructor but without it for others. Consider removing the `_` or adding it to other parameter names.
- N2 - line 41 - `EVENTS_SIGNER_ROLE` is not being used.
- N3 - line 48 - `lastStatusHash` is not being used. Why is it used at setProjectURI?. Consider removing it.
- N4 - line 66 - `fuulFactoryInstance` missing visibility.
- N5 - line 296 - Unnecessary emission of an empty array. Consider having different methods and/or events for depositing ERC721 and ERC1155.
- N6 - line 354 - Lock project funds forever if `removePeriodsEnds` is `0`. This variable is the `projectRemoveBudgetPeriod` from *FuulFactory.sol*.
- N7 - line 569 - Question in the code as a comment.
- N8 - line 572 - `amountToPartner + amountToEndUser` is the same as `totalAmount`.
- N9 - line 863 - Consider doing the check of is `isERC20` when the token is allowed by adding properties to the struct to reduce extra calls. Calling contract methods to

check compliance consumes a lot of gas and can be easily bypassed if a general contract with those methods is created.

Ignacio Mazzara - 27th May 2023.