G₀ Group

Security Review of

Gelato

June 17, 2020

Gelato / June 2020

Files in scope

Following solidity files in this repository:

https://github.com/gelatodigital/gelato-V1/commit/a6a263ab6850bfc09bd4d79ad38ec276b082e934

```
contracts/
    gelato_core/*.sol
    gelato_actions/
        gnosis/*.sol
    gelato_conditions/gnosis/*.sol
    gelato_provider_modules/*.sol
    libraries/*.sol
    user_proxies/gelato_user_proxy/*.sol
```

After changes resulting from the audit, these files from this repository have been added to the scope:

https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

```
contracts/
    gelato_actions/
        provider/*.sol
        GelatoActionPipeline.sol
```

Current status

As of June 16th all raised issues have been fixed by the developer

Issues

1. Internal functions in GelatoUserProxy mistakenly set as external

type: security / severity: critical

`GelatoUserProxy.callAction` and `GelatoUserProxy.delegatecallAction` are supposed to be internal, instead they are `external` which effectively allows anyone to control the proxy contract.

status - fixed

Issue has been fixed and is no longer present in https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

2. Re-entrancy issue potentially allows relayers to replay user transactions

type: security / severity: critical

`taskReceiptHash[_TR.id]` in the `GelatoCore.exec` function is deleted only after the transaction has been executed, allowing a relayer that controls some contract called during the transaction to replay it multiple times. This can lead to duplicate token transactions and a host of other potential critical issues.

status - fixed

Issue has been fixed and is no longer present in https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

3. Underflow issue in GelatoCore.exec

type: unexpected behavior / severity: medium

`GelatoCore:line 270` `gasleft() - _internalGasRequirement` can underflow, because there's some gas consumed since `require(startGas > _internalGasRequirement, "GelatoCore.exec: Insufficient gas sent")`.

status - fixed

Issue has been fixed and is no longer present in https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

4. Underflow issue

type: usability / severity: minor

`GelatoCore.sol:line 166` sideffect of `providerCanExec` not being called on tasks submitted by the provider is that no `taskSpecGasPriceCeil` can be specified for these tasks.

status - fixed

Issue has been fixed and is no longer present in
https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

5. Improper calculation of consumed gas may lead to relayers being shortchanged on rewards

type: usability / severity: medium

In `_processProviderPayables()` safemath exception can happen if `startGas` is higher than `_gelatoMaxGas` because `estExecTxGas` is capped to `_gelatoMaxGas` (which is constant) and `gasleft()` can be arbitrarily high. Even if the safemath exception isn't triggered, a relayer providing more than `_gelatoMaxGas` will reduce their reward.

status - fixed

Issue has been fixed and is no longer present in
https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

6. Having only one global value for minimum gas relayers have to provide to user calls can limit usage

type: usability / severity: medium

All providers have to share one setting for minimum gas provided to their actions that is stored in `gelatoMaxGas`, some relayers might prefer to use a different value to reduce costs.

status - fixed

Issue has been partially addressed by allowing self-providers to set a custom`gasLimit` value, these changes are included in
https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

7. Missing call to status updating function allows users to misuse provider paid calls

type: security / severity: major

`ActionWithdrawBatchExchange.action` should call `FeeFinder.redeemCredit` to prevent free withdrawals.

status - fixed

Issue has been fixed and is no longer present in
https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

8. Underflow issue in ActionPlaceOrderBatchExchangePayFee

type: unexpected behavior / severity: medium

In `ActionPlaceOrderBatchExchangePayFee.action` , `_order.sellAmount` will underflow if `_order.sellAmount` is smaller than fee.

status - fixed

Issue has been fixed and is no longer present in https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b

9. Price manipulation can lead to fee inflation

type: security / severity: major

In `ActionPlaceOrderBatchExchangePayFee.action` it's potentially possible to inflate fee paid by the user by atomic exchange price manipulation.

status - fixed

Issue has been fixed and is no longer present in https://github.com/gelatodigital/gelato-V1/commit/252f8911d26cee17c4691d474ff9e7e2ecd60c4b