G₀ Group

Security Review of

Argent MakerV2 and other updates

May 4, 2020

Argent MakerV2 and other updates / April 2020

Files in Scope

```
contracts/
    modules/
        ApprovedTransfer.sol
        RecoveryManager.sol
        TransferManager.sol
        common/
            BaseTransfer.sol
            RelayerModuleV2.sol
        maker/
            MakerV2Base.sol
            MakerV2Invest.sol
            MakerV2Loan.sol
            MakerV2Manager.sol
    infrastructure/
        MakerRegistry.sol
```

Current Status

As of May 5th, 2020 all of the reported issues have been fixed by the developer.

Issues

1. Missing ownership verification allows attacker to draw DAI from different user's CDP

Type: security / Severity: critical

In `MakerV2Loan.addDebt` a `verifyLoanOwner(_wallet, _loanId)` check is missing so attacker can successfully provide `_loanId` of a CDP that belongs to another user.

status - fixed

Issue has been fixed and is no longer present in
https://github.com/argentlabs/argent-contracts/tree/a0b1d873dd530a5fc73459e19a907b8b0eea1cdb

2. Missing ownership verification allows attacker to give away different user's CDP

Type: security / Severity: critical

In `MakerV2Loan.giveVault` a `verifyLoanOwner(_wallet, _loanId)` check is missing so attacker can successfully provide `_loanId` of a CDP that belongs to another user.

status - fixed

Issue has been fixed and is no longer present in
https://github.com/argentlabs/argent-contracts/tree/a0b1d873dd530a5fc73459e19a907b8b0eea1cdb

3. A functionality for giving the contract ownership of existing CDPs can be used to create false CDP ownership records

Type: security / Severity: critical

An attacker can provide a `_loanId` of an existing CDP in the ownership of the contract to `MakerV2Loan.acquireLoan` function and using a custom wallet contract they can trick the contract into creating a record in the `loanIds` mapping that allows them to gain control of the CDP.

Issue has been fixed and is no longer present in
https://github.com/argentlabs/argent-contracts/tree/a1b1ecba615bb782fd0f23fc684cd019199f9efa

4. An incorrect check in TransferManager.approveTokenAndCallContract allows owner to bypass daily spending limit

Type: security / Severity: major

In `TransferManager.approveTokenAndCallContract` the `isWhitelisted` check should be on `_spender` not `_contract`, otherwise if `_contract` calls attacker's address somewhere, this can be use to bypass spending limit.

status - fixed

Issue has been fixed and is no longer present in https://github.com/argentlabs/argent-contracts/tree/a0b1d873dd530a5fc73459e19a907b8b0eea1cdb

5. Owner of MakerRegistry contract can frontrun user transactions to MakerV2Loan and redirect their funds to an incorrect destination

Type: centralisation / Severity: medium

`MakerRegistry` owner can redirect user funds by frontrunning their transactions and redirecting the funds somewhere else. There are two ways to mitigate this, one is to in some way ensure `_joinAdapter` of `MakerRegistry.addCollateral` is legitimate part of the Maker system. The other is to allow users to provide their own values for `gemJoin` and `collateral` to `MakerV2Loan.joinCollateral` and check that they matche values returned by the registry. The same should probably be done for `MakerV2Loan.openVault` and `ilk`.

status - fixed

Issue has been fixed and is no longer present in https://github.com/argentlabs/argent-contracts/tree/a0b1d873dd530a5fc73459e19a907b8b0eea1cdb

6. Existing ownership record can be overwritten by MakerV2Loan.acquireLoan or MakerV2Loan.migrateCdp

Type: usability / Severity: major

If user already has a CDP with the MakerV2Loan contract, this ownership record will be overwritten when either of `MakerV2Loan.acquireLoan` or `MakerV2Loan.migrateCdp` is called, resulting in a loss of access to the original CDP and funds contained.

Issue has been fixed and is no longer present in https://github.com/argentlabs/argent-contracts/tree/a1b1ecba615bb782fd0f23fc684cd019199f9efa