G₀ Group

Security Review of Argent Upgrades

November 29, 2019

Overview

G0 Group was engaged to perform a security review of smart contract updates for the Argent smart contract wallet. G0 Group was contracted for a twelve person-day effort to that end. This review was initially performed on https://github.com/argentlabs/argent-contracts/tree/34da49e6defded4cce9602ea30baa53ef99ae76a.

Files in Scope

All solidity files in the following repository:
https://github.com/argentlabs/argent-contracts/tree/34da49e6defded4cce9602ea30baa53ef99ae76a

In particular, there are three new contracts:

```
contracts/
    modules/
        common/
            BaseTransfer.sol
        MakerV2Manager.sol
        TransferManager.sol
```

And three contracts with considerable changes:

```
contracts/
    exchange/
        TokenPriceProvider.sol
    modules/
        ApprovedTransfer.sol
    upgrade/
        SimpleUpgrader.sol
```

Result Summary

During the course of this review, 3 issues and one note were discovered and addressed. One issue was discovered independently by the client, and its fix confirmation has been included for completeness. All issues have been remediated and no further issues were discovered in
https://github.com/argentlabs/argent-contracts/commit/2de467a04145f1257aabcf88d4f8fe39e4a26dbe

Issues

1. Spending limit in TransferManager can be exceeded

Type: *security* / ***Severity:*** *major*

`verifyRefund` is called before a relayed transaction is executed, and `refund` (which actually updates the spending accounting) afterwards; therefore, it's possible to exceed the daily limit by recursively executing multiple relayed transactions inside one transaction so that all `verifyRefund` calls occur before the spending accounting gets updated.

Fix Description:

This issue has been addressed by adding an additional check to the `verifyRefund` function and is no longer present in
https://github.com/argentlabs/argent-contracts/commit/2de467a04145f1257aabcf88d4f8fe39e4a26dbe

2. Unnecessary code duplication

Type: *code quality* / ***Severity:*** *minor*

The `OnlyWhenUnlocked` modifier is defined identically in almost all modules, moving it inside a commonly inherited contract might improve code clarity.

Fix Description:

This issue has been addressed by moving the modifier to `BaseModule` and is no longer present in
https://github.com/argentlabs/argent-contracts/commit/2de467a04145f1257aabcf88d4f8fe39e4a26dbe

3. Filtering standard ERC20 function names in TokenTransfer to prevent unaccounted transfers of value is insufficient

Type: *security* / ***Severity:*** *major*

```
**This issue was discovered independently by the client and is included for the sake of completeness.**
```

Some ERC20 contracts use non-standard functions to transfer tokens, such as `increaseAllowance`, this means that merely preventing calls of `approve` and `transfer` functions is insufficient to prevent transfer of all ERC20 tokens. This could possibly allow users to bypass daily spending limits in the `TransferManager` contract.

Fix Description:

This issue has been addressed by filtering calls using a price oracle database of known ERC20 contracts as a blacklist. Argent has also added the ability for users who have disabled their daily limit to bypass this blacklist. This is necessary to ensure Argent cannot arbitrarily block users' wallets from interacting with contracts/dapps by fraudulently adding them to the price oracle. This issue is no longer present https://github.com/argentlabs/argent-contracts/commit/2de467a04145f1257aabcf88d4f8fe39e4a26dbe

Additional Notes

This upgrade newly allows the manager of `TokenPriceProvider` to manually update prices; previously prices had to be sourced from Kyber. This enables significantly more gas efficient updates and allows Argent to include tokens that are not on Kyber in daily limits; however, this also allows the system manager (Argent) to bypass the daily spending limits of all wallets through fraudulent price updates.