G$_0$ Group

Security Review of
Argent Update

October 9, 2020

Argent / August 2020

Files in scope

All files in the following repository and folder

https://github.com/argentlabs/argent-contracts/tree/eb69d91a47577293c1e675756f83d73fc04879c6/contracts

Current status

As of October 9, all issues have been addressed by the developer

Issues

1. It’s possible to revoke non-existent guardian addresses corrupting the state of the GuardianStorage contract

Type: unexpected behavior / Severity: medium

`GuardianManager.revokeGuardian` will accept as `_guardian` not only addresses of guardian accounts or contracts, but also the contract’s owners due to how `GuardianUtils.isGuardian` works. This allows `GuardianStorage.revokeGuardian` to be called with an address that isn’t in the `configs[_wallet].guardians` array or the `configs[_wallet].info` mapping. This will lead to the guardian at `configs[_wallet].guardians[0]` to be partially removed. With the `configs[_wallet].info` mapping for it left behind. This means the guardian will no longer be returned in `GuardianStorage.getGuardians` function, but will still be recognised by `GuardianStorage.isGuardian`. Another side-effect is that if `GuardianStorage.revokeGuardian` is called in the future with the address of the partially removed guardian, it will end up being removed completely, but the guardian at its former index will be now partially removed instead.

status - fixed

The issue has been fixed and is no longer present in

https://github.com/argentlabs/argent-contracts/commit/0042bfafdbad5bbdab2e447ef4a78d5f69ad3138

Notes

2. Useless extension of Storage in TokenPriceStorage contract

Type: redundant code / Severity: minor

`TokenPriceStorage` contract extends `Storage` contract, but doesn't make use of any functionality contained within.

status - fixed

The issue has been fixed and is no longer present in

https://github.com/argentlabs/argent-contracts/commit/0042bfafdbad5bbdab2e447ef4a78d5f69ad3138

3. Inconsistent implementation of transfer functions in the ApprovedTransfer contract

Type: inconsistent implementation / Severity: minor

All transfer functions in `ApprovedTransfer` contract except `approveWethAndCallContract` reset daily spending counter. The exception is without any apparent reason.

status - fixed

The issue has been fixed and is no longer present in

https://github.com/argentlabs/argent-contracts/commit/0042bfafdbad5bbdab2e447ef4a78d5f69ad3138

4. Potentially dangerous uporotected public function in BaseModule contract

Type: fragile code / Severity: minor

`BaseModule` contract which is extended by all module contracts contains `recoverToken` function, which can be called by anyone and generates a `transfer(address, uint256)` call to an arbitrary address including other modules, storage contracts and contracts that contain assets owned by the modules. To my knowledge this isn't exploitable right now, but in case any contract in the mentioned categories implements a transfer function in the future, this could be used to call it maliciously.

status - acknowledged

BaseModule has been renamed to BaseFeature, but the function has remained. There are no immediate vulnerabilities resulting from this, but it should be noted that Features should never hold ERC20 balances, other than in a non-reentrant atomic call, until this function is removed.