$G_0$ Group

# **Security Review of** Argent Update

September 29, 2020

# Argent / September 2020

## Files in scope

All files in the following repository and folder

https://github.com/argentlabs/argent-contracts/tree/665448dc4c337c8f0153c372e42dff8b7b68d638/contracts

## Current status

All serious issues have been addressed by the developer

# Issues

## 1. If a feature that was part of a past wallet version but wasn't part of the last version is added to the wallet, it will be initialised again

*Type: unexpected behavior / Severity: minor*

In `VersionManager.upgradeWallet` the condition for calling `init()` on a feature contract that is part of the new version is `fromVersion == 0 || !isFeatureInVersion[feature][fromVersion]` this mean that it's called if either this is the first version that the wallet is being upgraded to, or if the feature wasn't part of the last version the wallet used. This however means that if the feature was part of one of the older versions the wallet used, the `init()` function will be called anyway. This might lead to unexpected and problematic behavior.

***status - issue has addressed by an addition of a warning in the function documentation to avoid the problematic setup***

## 2. An issue in VersionManager allows administrator to take control of all wallets

*Type: security / Severity: major*

In `VersionManager.invokeStorage` doesn't check if the `_wallet` argument (which is used to verify that the wallet has authorised caller to control its storage) matches the wallet address contained in the `_data` argument, that is used to determine the wallet for which the storage will be actually updated. In result any caller that can update storage for any wallet can update it for all wallets. This opens multiple ways for the owner of the `VersionManager` contract to gain control of all wallets that use the `VersionManager` contract.

***status - issue has been fixed and is no longer present in***

https://github.com/argentlabs/argent-contracts/tree/2956f8d91e41887cac5cfba584d87ea5ceca38d7/contracts

## 3. The fact that the manager of versionManager can change the minimal wallet version at any time makes behavior of counterfactual wallets non-deterministic

*Type: security / Severity: medium*

Advantage of wallets deployed through `WalletFactory.createCounterfactualWallet` is that the code and security related settings of the wallet determine the address that the wallet will reside on. This allows users to treat the address as their own even before the wallet has been deployed knowing that they will eventually gain control of all assets owned by the address. This expectation is partially broken by the fact that the `_version` argument of `WalletFactory.createCounterfactualWallet` can be overriden in `VersionManager.upgradeWallet` by the `minVersion` setting that can be changed at any time by the owner of the `VersionManager` contract (Argent multisig). This opens a hypothetical attack that allows the `VersionManager` owner to steal assets that have been transferred to the address of a counterfactual contract before it has been deployed. Since only a small and very specific set of users can be affected by the attack at any time, which limits the potential payoff and since the attack would irreversibly tarnish the reputation of the Argent organisation, it's very unlikely to be commited even by a purely self-interested agent.

***status - the issue has been acknowladged by the developer***