ANT Controller

Smart contract source code audit

Prepared for Aragon - August 2020

coinspect

Smart Contract Source Code Audit - ANTController

Prepared for Aragon • August 2020

v200827

1. Table Of Contents

2. Executive Summary

In August 2020, Aragon engaged Coinspect to perform a source code review of the final Aragon Network Token Controller contract. The objective of the audit was to evaluate the security of the smart contract and verify that the deployed contract matches the reviewed source code.

The assessment covered the `master` branch of the repository at https://github.com/aragon/aragon-network-token/tree/master/packages/controller up to (and including) commit `ece3c1b7bee846d95bea05177373f28b7dc3af99` of **August 20th**.

No issues were identified during the assessment.

3. Introduction

The Aragon Network Token is an ERC20 contract with additional functionality. The extra functionality is provided through a *controller* contract, and it includes arbitrary transfers, enabling/disabling transfers, and minting.

The focus of the audit was a new controller contract that greatly restricts the extra functionality, effectively limiting the power that a single entity (in particular the community multisig) could have over the token. The new controller contract is final, i.e. it cannot be replaced, providing in this way an assurance to the users that the tokens cannot be tampered with.

The audit started on August 24th and was conducted on the `master` branch of the Git repository at https://github.com/aragon/aragon-network-token/tree/master/packages/controller up to and including commit `ece3c1b` of **August 20th:**

```
commit ece3c1b7bee846d95bea05177373f28b7dc3af99
Author: Brett Sun <...@gmail.com>
Date:   Thu Aug 20 10:30:33 2020 +0200

    controller: update mint() interface to generateTokens() (#17)
```

The scope of the audit was limited to the following Solidity source files, shown here with their sha256sum hash as of commit `ece3c1b`:

```
46acec4437623d5857fccbf36e4ab59f42df1a9fb2f5588bc645e672a00b6ef3 ANTController.sol
8dacae815fbecbeef922fee11c0c0958f759a9b125e1ee2afb03013398a15ae9 IMiniMeLike.sol
faaa70d64b3c455c1f241fc8c0308d0719ef4171e722cf5f2df65ce67bb3309f ITokenController.sol
```

4. Assessment

The `ANTController` contract is very simple, and it is very clear that it behaves as expected. This controller is final: **once it is set as controller of the ANT contract it cannot be changed**, because only the controller can set a new controller and functionality for this is not provided in the `ANTController` contract. The new `ANTController` contract disables forever some of the extra features supported by the ANT contract in addition to the ERC20 standard. The disabled functionality includes:

- Arbitrary `transferFrom` bypassing allowance;
- Enabling/disabling transfers by calling `enableTransfers`;
- Proxy payments (sending ether to the ANT contract);
- Callbacks on transfers and approvals

The ANT contract originally allows the controller to perform arbitrary transfers and temporarily disable transfers, and the fact that this functionality is not implemented in the `ANTController` contract guarantees that arbitrary transfers and transfer disabling will never be used. Also, the function `proxyPayment` in the new `ANTController` contract returns always `false`, rejecting any ether sent to the ANT contract. The `onTransfer` and `onApprove` callbacks are ignored (they are implemented just returning `true`), and no other code is executed on transfers or approvals.

The only controller functionality allowed by the new `ANTController` contract is minting. It is implemented in such a way that a minter address is set when the contract is created, and only this address can call the function `generateTokens` to mint new tokens. Also, the minter address is allowed to change the minter to a new address.

The repository contains 29 unit tests for the smart contract. Besides unit tests, the repository also includes a suite of 15 E2E tests that run against a fork of mainnet, to ensure that the new `ANTController` works correctly with the ANT contract already deployed in Ethereum mainnet. All unit tests and E2E tests pass without problems.

It is important to mention that, since after setting the new `ANTController` it will no longer be possible to call `enableTransfers`, if transfers are disabled at the time of setting the new controller the ANT contract will be left unusable. So **it is important to make sure that transfers are enabled before setting the new controller**.

5. Post-Deployment Verification

The contract source code is verified in Etherscan, the leading Ethereum explorer. Coinspect independently verified that the deployed bytecode matches the source code reviewed. To do this, Coinspect compiled the contract using exactly the same Solidity version and parameters that were used in the deployed version, and removed the metadata before comparing the resulting bytecode with the deployed one. The differences between the locally compiled bytecode and the one stored in the Ethereum blockchain are the metadata added by Solidity and the constructor parameters. The constructor parameters are as expected.

```
Solidity version 0.5.17+commit.d19bba13 optimizer enabled, 10000 runs.
EVM: istanbul
```

The `ANTController` contract was deployed at address **`0x2443d44325bb07861Cd8C9C8Ba1569b6c39D9d95`** by transaction `0xdd73a27460f8bcb60ab4566ec60400fd5a63a5d2a22fc36e9bf7e85f27e1e1fa` (Etherscan) from address `0xe04cabcb24e11620dd62bb99c396e76ceb578914` in August 20th (mined at 2020-08-20 20:58:20 +0000 block #10699390).

The parameters for the constructor (`IMiniMeLike _ant, address _minter`) were the following addresses:

The ANT contract: `0x960b236a07cf122663c4303350609a66a7b288c0`
The multisig wallet that is set as minter: `0xbeefbeef03c7e5a1c29e0aa675f8e16aee0a5fad`

Coinspect verified the state changes made by the init code were only the expected ones to set the minter and ANT token addresses.
The contract receipts for block 10,699,390 show the expected results including the event `ChangedMinter (0xbeefbeef03c7e5a1c29e0aa675f8e16aee0a5fad)` being emitted.

The metadata includes a Swarm content hash that was not retrieved from the Swarm network but is not important to verify the integrity of the contact. Decoded metadata:

```
{"bzzr1":"6d64f7ca8dc75e204cb471ac35210abb6b890af8b172b120f6468a7657e6ecac
","solc": "000511"}
```

6. Conclusion

The contract is simple, clear, and will effectively limit the power of any single entity (in particular the community multisig wallet) on the token. It is well tested, both with unit tests and also with E2E tests on a fork of mainnet.

Since setting the new controller will be final, it is important to do whatever possible to avoid mishaps. In particular, the new controller won't allow calls to `enableTransfers` any more, and if transfers were disabled when setting the new controller they would remain disabled forever. It is recommended to make sure that there is no chance of transfers being disabled at the time of setting the new controller address in the ANT contract.

7. Disclaimer

The present security audit does not cover the endpoint systems and wallets that communicate with the contracts, nor the general operational security of the company whose contracts have been audited. This document should not be read as investment advice or an offering of tokens.