(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/)

**SMART CONTRACT AUDIT (HTTPS://BLOG.COINFABRIK.COM/CATEGORY/SMART-CONTRACTS/SMART-CONTRACT-AUDIT-SMART-CONTRACTS/)**

# DMToken Security Audit V2 2019 Updated

Ariel Yabo (https://blog.coinfabrik.com/author/ariel-yabo/)

February 25, 2019 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dmtoken-security-audit-v2-2019-updated/)

Contents

# Introduction

CoinFabrik has been hired to audit the DMToken smart contract. First of all, we will describe what is the analysis performed by our team. Next, we will provide a summary of our discoveries and we will detail all our findings. And at the end, we will express our opinion in the conclusion.

# Summary

The contract audited is at address 0x2ccbff3a042c68716ed2a2cb0c544a9f1d1935e1 (https://etherscan.io/address/0x2ccbff3a042c68716ed2a2cb0c544a9f1d1935e1#code). The contract was deployed at transaction 0x111287826ad158a7ae11a9893f03e9e8998ed7b2bac061c83bed7b68e1958a6d (https://etherscan.io/tx/0x111287826ad158a7ae11a9893f03e9e8998ed7b2bac061c83bed7b68e1958a6d).

# Description

- Contracts name: DMToken
- Compiler: version 0.4.18 with optimization enabled

The contract is a simple ERC20. It provides additional functionality that allows some users the vesting of token gradually over time.

# Analysis performed

- Misuse of the different call methods: call.value(), send() and transfer().
- Integer rounding errors, overflow, underflow and related usage of SafeMath functions.
- Old compiler version pragmas.
- Race conditions such as reentrancy attacks or front running.
- Misuse of block timestamps, assuming anything other than them being strictly increasing.
- Contract softlocking attacks (DoS).
- Potential gas cost of functions being over the gas limit.
- Missing function qualifiers and their misuse.
- Fallback functions with a higher gas cost than the one that a transfer or send call allows.
- Fraudulent or erroneous code.
- Code and contract interaction complexity.
- Wrong or missing error handling.
- Overuse of transfers in a single transaction instead of using withdrawal patterns.
- Insufficient analysis of the function input requirements.

# Detailed findings

## Critical severity

No issue has been found.

## Medium severity

No issue has been found.

## Minor severity

No issue has been found.

# Observations

## Old compiler version

The audited contracts were deployed with version 0.4.18 of the solidity compiler (solc). At the time of this audit the latest release of solc is 0.5.4. We checked vulnerabilities known to affect this old version of the compiler against this particular contract. We found that none of them compromise the contract.

They consist of two high-medium severity bugs:

- ExpExponentCleanup: Only the exponentiation operation with types smaller than 256 bits is impacted. Some of the higher bits are not zeroed before use resulting in difficult to predict behaviour. **We didn't find any use of exponentiation operations in this contract**.
- NestedArrayFunctionCallDecoder: Functions calls where the return type is a multidimensional fixed size array are impacted. We found no function using such a return type in this contract.

There are several other changes to Solidity language since version 0.4.18 of solc (October 2017). These changes should not affect the functionality of the contract at all.

- Use of *view* and *pure* modifiers instead of *constant*.
- Use of *emit* to disambiguate generating events of function calls.
- Introduction of the *constructor* keyword.
- A slightly richer type system.

# Conclusion

The contract is a simple ERC20 token with additional functionality that the contract owner might allow some users to vest tokens gradually over time.

We found no issue with the currently deployed contracts. There are some vulnerabilities in the used solidity compiler, but we found no use of the affected operations in the smart contracts.

**Disclaimer: This audit report is not a security warranty, investment advice, or an approval of DMToken since Coinfabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.**

## Related Posts

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/beluga-pay-bbi-security-audit/)

Beluga Pay (BBI) Security Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/beluga-pay-bbi-security-audit/)

Coinfabrik has been hired to audit the smart contracts which were included in the BBI...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/cryptosolartech-security-audit/)

Cryptosolartech Security Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/cryptosolartech-security-audit/)

Coinfabrik's smart contract audit's team was asked to audit the contracts for the Cryptosolartech sale.…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

RCN Smart Contracts Audit v2 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

The smart contracts that have been audited were taken from the RCN repository at: https://github.com/ripio/rcn-network/tree/v2.…

(https://blog.coinfabrik.com/smart-contracts/etherparty-token-smart-contract-security-audit-coinfabrik/)

EtherParty Smart Contract Security Audit (https://blog.coinfabrik.com/smart-contracts/etherparty-token-smart-contract-security-audit-coinfabrik/)

Coinfabrik team has been hired to audit Etherparty smart contracts. Firstly, we will provide a…

Tags:

audit (https://blog.coinfabrik.com/tag/audit/)

security (https://blog.coinfabrik.com/tag/security/)

SHARE ON

(https://www.facebook.com/sharer/sharer.php?u=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dmtoken-security-audit-v2-2019-updated/)

(https://twitter.com/intent/tweet?text=DMToken%20Security%20Audit%20V2%202019%20Updated&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dmtoken-security-audit-v2-2019-updated/)

(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-

smart contract audits (https://blog.coinfabrik.com/tag/smart-contract-audits/)

(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-content/uploads/2017/12/dmtoken.png&description=DMToken+Security+Audit+V2+2019+Updated)
(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dmtoken-security-audit-v2-2019-updated/&title=DMToken%20Security%20Audit%20V2%202019%20Updated&source=CoinFabrik%20Blog)

← PREVIOUS ARTICLE

**Decentraland Security Audit (https://blog.coinfabrik.com/uncategorized/decentraland-security-audit/)**

NEXT ARTICLE →

**Dogethereum State (https://blog.coinfabrik.com/uncategorized/dogethereum-state/)**

## You may also like

(https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

**Magic Bridge Audit (https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)**

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

2 months ago

Smart Contract Audit (https://blog.coinfabrik.com/category/smart-contracts/smart-contract-audit-smart-contracts/)

**MintingFactoryV2, BaseUpgradableMarketplace & KODAV3UpgradableGatedMarketplace (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)**

(https://www.youtube.com/channel/UC2GmjCr7aEz-il31kqOy9aw)

(https://www.facebook.com/CoinFabrik/)

(https://www.reddit.com/r/CoinFabrik/)

(https://github.com/coinfabrik)

Made with ♥ in Buenos Aires, Argentina.