QuillAudits

Audit Report
April, 2021

AFRICA TO MARS

Contents

Scope of Audit

The scope of this audit was to analyse and document ATM smart contract codebase for quality, security, and correctness.

Check Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis
In this step we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.
SmartCheck.

Static Analysis
Static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test security of smart contracts.

Code Review / Manual Analysis
Manual Analysis or review of code was done to identify new vulnerability or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of automated analysis were manually verified.

Gas Consumption
In this step we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit
Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Slither, SmartCheck.

Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity and each of them has been explained below.

High severity issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality and we recommend these issues to be fixed before moving to a live environment.

Medium level severity issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems and they should still be fixed.

Low level severity issues

Low level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity four issues which indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of issues per severity

	High	Medium	Low	Informational
Open	0	0	2	1
Closed	0	0	0	0

Introduction

During the period of March 26th, 2021 to April 28th, 2021 - Quillhash Team performed a security audit for ATM smart contracts.
The code for the audit was taken from following the official Etherscan link:
https://etherscan.io/address/0x9b91ef0d78488c5ef4c509eb7a73f7d8ca650ce4#code

Issues Found – Code Review / Manual Testing

High severity issues

Not Found

Medium severity issues

Not Found

Low level severity issues

1. **Absence of Error messages in Require Statements**

   Description
   No require statement in the ATM.sol contract includes an error message. While this makes it troublesome to detect the reason behind a particular function revert, it also reduces the readability of the code.

   Recommendation:
   Include error messages in every require statement.

2. **External Visibility should be preferred**

   Description
   Those functions that are never called throughout the contract should be marked as external visibility instead of public visibility.
   This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as external within the contract:

- allowance
- transfer #Line 185
- increaseAllowance
- decreaseAllowance
- transferFrom
- approve

Recommendation:
If the PUBLIC visibility of the above-mentioned functions is not intended, then they should be assigned the EXTERNAL visibility keyword.

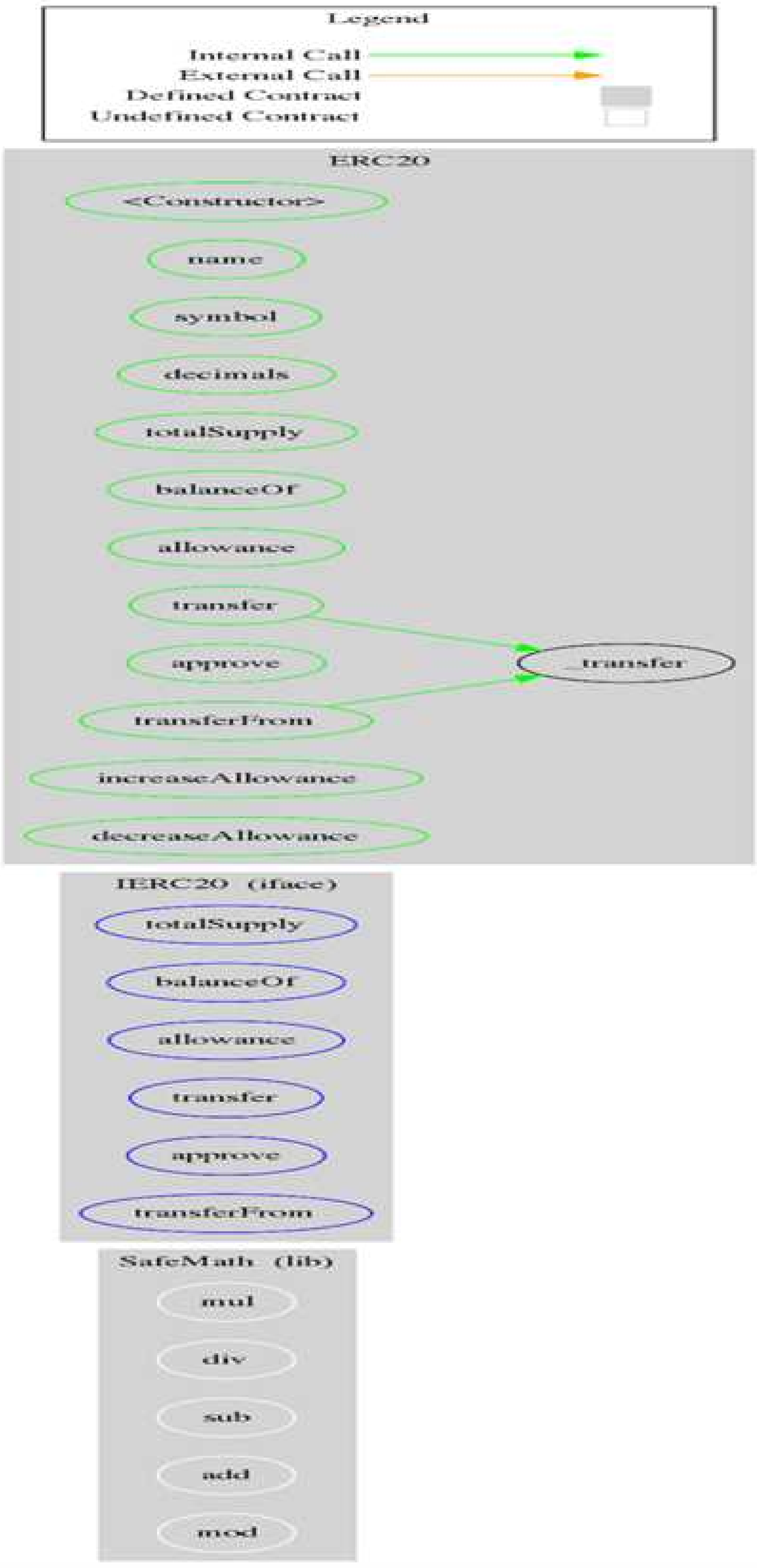
Informational

3. Token Contract doesn't include "mint" and "burn" functionalities

Description
Although the ATM.sol token contract contains all the imperative functionalities of ERC20 token standard, it doesn't include the mint and burn functions that are of utmost importance in case of minting new tokens or burning existing tokens.

Is this intended?
Recommendation:
This does not affect the intended behaviour of the contract and is not necessary if the total supply of the token is supposed to be constant in the future.
However, if new ATM tokens are to be minted or existing tokens are supposed to be burnt in future, the mint and burn functionality should be implemented in the smart contract.

Contract Control Flow



Closing Summary

Overall, smart contracts are very well written and adhere to guidelines. During the process of audit, No issues of high or medium severity were found. However, some low severity issues were found, which might affect the intended behaviour of the contract and they have been documented above.

Moreover, No instances of Re-entrancy or Back-Door Entry were found in the contract.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the ATM platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the ATM Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

AFRICA TO MARS

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

hello@quillhash.com