QMALL Smart Contracts Audit

January, 2022

Yaroslav Tumanov, Oleksandr Zadorozhnyi

1. **INTRODUCTION.** QMALL requested 4irelabs a review of the contracts implementing their token and vesting.

   There are 2 contracts in scope (and their parents):

   - Token https://gitlab.com/qMall-ex/smart-contracts/-/blob/main/Token
   - TokenVesting https://gitlab.com/qMall-ex/smart-contracts/-/blob/main/Vesting

2. **WARRANTY**. Audit is provided on an "as is" basis, without warranty of any kind, express or implied. The Auditor does not guarantee that the Code Review will identify all instances of security vulnerabilities or other related issues.

3. **EXECUTIVE SUMMARY**. No critical issues were found in the contracts. QMALL token contract is compliant with the finalized ERC20 standard. New tokens could not be minted. There are **no** known compiler bugs, for the specified compiler version, that might affect the contracts logic.

4. **CRITICAL BUGS AND VULNERABILITIES**. No critical issues were found in the contracts.

5. LINE BY LINE REVIEW.

5.1. Token

5.1.1. **Resolved.** Line 93 and next lines. Note: Typos “Blok..” instead of “Block..”.

5.1.2. **Resolved.** Line 95. Minor: Mapping visibility should be updated to private.

5.1.3. **Resolved.** Lines 136,145 Bloklist/unBlocklist Note: Bloklister is able to set blocked status if user already blocked (and vise versa). In this case, Blocklisted event will be emitted 2 times. Possible solution: check current user status and if it's blocked, then do not allow to execute block function again and vise versa.

5.1.4. Line 136. Major: Privilege of blocklister to block any user token balance directly. Recommendation: if blocklister block function was not implemented by design then it should be restricted.

5.1.5. **Resolved.** Line 160. Note: Inheritance from **Ownable** contract is not needed, because Bloklister base contract has this.

5.1.6. **Resolved.** Line 168 Note(gas optimization): variable **_decimals** could be set as immutable and assigned only from constructor (immutable is similar to the constant)

5.1.7. **Resolved.** Line 171. Note(gas optimization): **_totalSupply** has default value 0, no need to set 0 and spend gas for that.

5.1.8. **Resolved.** Line 171. Minor: Variable visibility should be updated to private.

5.1.9. Resolved. Line 173. Note: **1_00_000_000e18** better for reading as **100_000_000e18**

5.1.10. Resolved. Line 175. Minor: Token name has additional space in beginning.

5.1.11. Resolved. Line 178, 183, 213, 216, 241, 254, 265, 325. Minor: Should use **_msgSender()** instead of **msg.sender**.

5.1.12. Resolved. Line 182. Major: Privilege of the owner to mint any amount of token. Recommendation: if the owner's minting function was not implemented by design then it should be restricted.

5.1.13. Resolved. Line 183. Note(gas optimization): If **mint()** have **onlyOwner** modifier - there is no need to add **notBlocklisted** modifier.

5.1.14. Resolved. Line 220. Note(gas optimization): Function **_mint()** returns bool value which was never processed. *Possible solution:* Remove returning value in function **_mint()**

5.1.15. Resolved. Line 254. Minor: There are no need for blocklisting modifiers since function **transferFrom()** checks all addresses for blocklist. But if blocklisting for approval is required, then blocklisting modifiers should also be added to **increaseAllowance()** and **decreaseAllowance()** functions.

5.1.16. Line 325. Minor: Should use **SafeERC20** library during recovering tokens to be sure that transfer is successful

5.1.17. Resolved. Lines 327, 332. Note(gas optimization): Not needed functions since they are empty.

5.1.18. Resolved. Line 213, 241, 254, 280, 290. Note(gas optimization): Visibility should be updated to external for better gas usage.

5.2. Vesting

5.2.1. Resolved. Line 560. Critical: require statement expect that revoked value is true, but the logic of modifier onlyIfVestingScheduleNotRevoked() expect only not revoked vesting schedules. Possible solution: update require statement to next:

require(!vestingSchedules[vestingScheduleId].revoked);

QMALL Smart contracts Audit Update

January, 2022

Yaroslav Tumanov, Oleksandr Zadorozhnyi

Contracts have been updated and Critical/Major (5.2.1, 5.1.12) issues have been resolved. Tokens could be claimed if not revoked on Vesting contract. Token owner is not able to mint additional QMALL tokens.

In Addition other issues have been fixed in the token contract — 5.1.5, 5.1.7, 5.1.9, 5.1.11, 5.1.12, 5.1.13, 5.1.14, 5.1.15, 5.1.17, 5.1.18, 5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.8, 5.1.10