RealFevr Marketplace Security Analysis

by Pessimistic

This report is public.

Published: September 8, 2021

Abstract

In this report, we consider the security of smart contracts of RealFevr Marketplace project. Our task is to find and describe security issues in the smart contracts of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, security audit is not an investment advice.

Summary

In this report, we considered the security of RealFevr Marketplace smart contracts. We performed our audit according to the procedure described below.

The initial audit showed several issues of medium severity, including ERC20 standard violation and Overpowered owner. Also, a number of low severity issues were found.

The project has no documentation. The audited part of the project is not covered with tests.

After the initial audit, the code base was updated. In this update, no fixes were applied, but new functionality was added. The recheck revealed two new issues in the updated code.

General recommendations

We recommend fixing the mentioned issues, adding documentation to the project, adding tests, adding CI to run tests, calculate code coverage, and analyze code with linters and security tools. We also recommend limiting the powers of the owner and following CEI pattern.

Project overview

Project description

For the audit, we were provided with RealFevr project on a public GitHub repository.

The scope of the audit only included:

- **MarketplaceRealFvr.sol** file and its dependencies on commit f3a1372a82feb62751c9fb125ccecc997344f7d2.
- **OpenerRealFvr.sol** file and its dependencies on commit 1d2d29a666898f318e62f65a6e3d86928d6fc55f.

The project has no documentation, the code has no NatSpecs. `npm install` command fails.

The audited code is not covered with tests.

Code base update

After the initial audit, the code base was updated. For the recheck, we were provided with commit 2cfda872d107eb0fc3ea01ed3748add918f11683.

Procedure

In our audit, we consider the following crucial features of the code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices.

We perform our audit according to the following procedure:

- Automated analysis
  - We scan project's code base with automated tools: Slither and SmartCheck.
  - We manually verify (reject or confirm) all the issues found by tools.
- Manual audit
  - We manually analyze code base for security vulnerabilities.
  - We assess overall project structure and quality.
- Report
  - We reflect all the gathered information in the report.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger smart contracts security. We highly recommend fixing them.

The audit showed no critical issues.

Medium severity issues

Medium issues can influence project operation in current implementation. We highly recommend addressing them.

ERC20 standard violation

EIP-20 states:

```
Callers MUST handle false from returns (bool success). Callers MUST NOT
assume that false is never returned!
```

However, the returned value of `transferFrom()` call is not checked at line 65 of **OpenerRealFvr** contract.

Overpowered owner

The owner of **ERC721Marketplace** contract can change `erc20Address` and `erc721Address` values at any moment without notifying users. As a result, the users of the project can get results they do not expect when purchasing or selling non-fungible tokens.

The owner can also change `feeAddress` and `feePercentage` values.

After the code base update, the owner can also censor sales of NFTs using `removeERC721FromSaleAdmin()` function.

The owner of **OpenerRealFvr** contract can lock the contract, change purchase token address, and set token price in USD; create, offer, and delete packs, set token URI and base URI.

In the current implementation, the system depends heavily on the owner of the contract. Thus, there are scenarios that may lead to undesirable consequences for the users, e.g. if the owner's private keys become compromised.

We recommend designing contracts in a trustless manner or implementing proper key management, e.g. multisig.

> *Comment from developers: the idea is not not to censor sales, once they can put them on sale again, the ideia of this function is to have a mechanism to cancel all the sales in order to facilitate an upgrade to a new smart contract when we need, and this way, we avoid at least the cancel fees for the users. If we had this function before, the bug we had on the V1 of the smart contract would be easier to solve.*

No documentation

The project has no documentation. As a result, it is sometimes unclear what the intention of the code is, and whether its behavior is correct, and the architecture of the project is appropriate.

Proper documentation should explicitly describe the purpose and behavior of the contracts, their interactions, and main design choices. It is also essential for any further integrations.

No tests

The project has a few tests. However, the audited code is not covered. Testing is crucial for code security and audit does not replace tests in any way.

We highly recommend both covering the code with tests and making sure that the test coverage is sufficient.

Low severity issues

Low severity issues can influence project operation in future versions of code. We recommend taking them into account.

Code quality

- CEI pattern is violated all over the code. We highly recommend following CEI pattern since it does not increase the cost of execution for honest users, but improves usage predictability.
- In **ERC721MarketPlace** contract, variables `salesById` and `saleIncrementId` at lines 57 and 60 are unused. Consider removing them. For off-chain operations, consider emitting events instead.
- Consider emitting an event on price change in `setTokenPriceInUSD()` function of **OpenerRealFvr** contract.
- The name of `_openedPacks` variable is misleading since it increments when a pack is bought or offered but remains unchanged when a pack is opened.
- Consider rewriting `if-else` block at lines 32–83 of **MarketplaceRealFvr** contract to minimize code duplication.
- Consider declaring functions as `external` instead of `public` where possible.
- Typo at line 241 of **OpenerRealFvr** contract: `be` is missed in `require` message.

Gas consumption

- In `removeERC721FromSale()` function of **ERC721Marketplace** contract, the check of token id is redundant, since the check of the seller at line 64 confirms that NFT is for sale.
- In `buyERC721()` function of **ERC721Marketplace** contract, consider saving `sales[_tokenId]` value to a local variable to reduce gas consumption.
- Variable `pack` is unused in `offerPack()` function of **OpenerRealFvr** contract.
- In `mint()` function of **OpenerRealFvr** contract, the check at line 229 is redundant since it is already implemented in `ERC721._mint()`.
- When iterating through an array from storage, consider saving its length to a local variable, since reading `.length` property on each iteration is expensive, e.g. at line 124 of **OpenerRealFvr** contract.
- (new) Fields `canceled` and `sold` of `Sale` struct in **ERC721Marketplace** contract are redundant, since elements of `sales` mapping are deleted when sale is completed or removed.

Dependency management

- **package-lock.json** file is in **.gitignore**. Best practice is to commit **package-lock.json** to the repo.
- Scripts in **package.json** file require `truffle`, `eslint`, `babel-node`, `babel`, `rimraf`, `jsdoc`, `ganache-cli`, and `onchange` packages to be installed globally.
- Tools should be in `devDependencies` section rather than in `dependencies` of **package.json** file.
- `npm install` command fails for the project on commit 1d2d29a666898f318e62f65a6e3d86928d6fc55f due to missing dependencies.

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Daria Korepanova, Security Engineer
Vladimir Tarasov, Security Engineer
Boris Nikashin, Analyst
Irina Vikhareva, Project Manager

September 8, 2021