Guard

Smart contracts security assessment

Final report

Tariff: Standard

Cool Dogs Club

May 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for Cool Dogs Club team.

Name	Cool Dogs Club
Audit date	2022-05-17 - 2022-05-17
Language	Rust
Platform	NEAR

Contracts checked

Name	Address
NFT	
LoanFactory	

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Math logic - FIXED (LoanFactory)

Users' deposits can be used as a source of funds for rewards. The amount of rewards depends on time, not on commission amounts. Because of that, this can happen:

1. Alice and Bob each deposit 1 NEAR token into the contract through the `loan_deposit()` function.
2. Bob waits, claims his rewards, for example, in the amount of 0.1 NEAR, and withdraws his deposit in the amount of 1 NEAR.
3. Alice can't withdraw her deposit because the contract holds less than 1 NEAR

Also, the contract doesn't take into account the commissions from nft loans.

Recommendation:

1. It will be better to make the calculation of rewards proportional to commission amounts, not to the time.

2. The contract should add the commission amount to the `total_balance` variable.
3. Rewards should be separated from users' deposits.

Response: Following smart contracts recheck, Cool Dogs Club team fixed the finding.

2. Math underflow (LoanFactory)

Function `internal_decrease_balance` updates user's shares to `diff.amount - old.amount`, which usually underflows.

```
  pub(crate) fn internal_decrease_balance(&mut self, account_id: &AccountId, amount:
&U128) {
    let current = self.accounts.get(&account_id).unwrap_or_else(|| 0);
    let num_shares = U128::from(self.total_shares.0 * amount.0 / self.total_balance.0);
    let current_shares = self.shares_by_account.get(&account_id).unwrap_or_else(||
U128::from(0));
    let new_shares = U128::from(num_shares.0 - current_shares.0);

    if amount.0 > current {
        env::panic_str("No funds");
    }

    let next = current - amount.0;

    self.accounts.insert(&account_id, &next);
    self.total_balance = U128::from(self.total_balance.0 - amount.0);

    self.total_shares = U128::from(self.total_shares.0 - num_shares.0);
    self.shares_by_account.insert(&account_id, &new_shares);
  }
```

Recommendation: Fix the math.

Medium severity issues

1. Native transfers - FIXED (LoanFactory)

In the functions `loan_nft_pay()`, `loan_resolve_nft()`, `loan_deposit()`, `loan_withdraw()`, `loan_withdraw_all()`, `loan_claim_rewards()` the results of transfers of the native token aren't checked. It is a better practice to check all Promise results.

Response: Following smart contracts recheck, Cool Dogs Club team fixed the finding.

2. Rent (LoanFactory)

Users' deposits shouldn't be used as a rent for contract's storage.

Recommendation: The contract can implement a storage management or the rent can be payed by an admin.

Low severity issues

1. Arithmetic errors - FIXED (LoanFactory)

In the functions `LoanFactoryResolver::loan_resolve_nft()` and `LoanFactory::internal_reward_unclaimed_of()` there are calculations where division is made before multiplication. It is better to do all the multiplactions first, and then all the divisions.

Response: Following smart contracts recheck, Cool Dogs Club team fixed the finding.

2. Additional assert (LoanFactory)

In function `loan_nft()` it is required to add checking `assert_one_yocto()` from near_sdk in the beginning because further in function there is a function call to another contract that attaches value of one yocto.

3. Function access (LoanFactory)

Functions `loan_resolve_nft()` and `loan_resolve_nft_claim()` should be accessible only for the contract itself.

Conclusion

Cool Dogs Club NFT, LoanFactory contracts were audited. 2 high, 2 medium, 3 low severity issues were found.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

0x Guard