TRC404 Smart Contract Audit Report

Sun Apr 28 2024

contact@scalebit.xyz https://twitter.com/tonbit_

TonBit

TRC404 Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	TRC-404 is an experimental, mixed Jetton / NFT implementation with native liquidity and fractionalization for semi-fungible tokens.
Type	Token
Auditors	TonBit
Timeline	Tue Apr 23 2024 - Sun Apr 28 2024
Languages	FunC
Platform	Ton
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/NotFoundLabs/TRC-404
Commits	456bc981c67d06c48d112709cc11dafd225bfdc8 79bdb2b2bc7b03e528de2153e3afc89aa044f4b8 8a2b83f48c6a3311131854d91196ce7ef2d7a02f

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
T4M	sources/trc404_master.fc	53a1aab90ee6b676bc393b862e46cb554073d2e1
SME	sources/utils/sendMessage.fc	397bcad878509f326f118c05e1fa8efb4e814ea2
STD	sources/utils/stdlib.fc	97609508caf6987acda0940f60d4ebbb45413c63
CHA	sources/utils/chain.fc	3e86ce82bee70992c9b0f7b4fcacf0cacfcfec1b
CON	sources/utils/const.fc	dad100f85a2c9ffd1d7ff718331dd480e101ae5e
T4W	sources/trc404_wallet.fc	1fb8864771bd45da8be8ea51a07caa5b420f3a80
CME	sources/message/common_message.fc	52bc8cefb30011442951225b08a89698c7f2d725
MME	sources/message/master_message.fc	51baf65669f72df9b5d6eca600eeb9110514940b
NCM	sources/message/nftCollection_message.fc	b633e28a390f3e21bb77711ff359a37dbcaf9336
NIM	sources/message/nftItem_message.fc	27290884500cdf97f6970f85ad05d3e8802bf5ee
WME	sources/message/wallet_message.fc	b330deea75b16a65305bc4940d9e97d1afb914ea

T4NC	sources/trc404_nftCollection.fc	c777c57af9b1884051598c37256f5dc5c2cccd1c
T4NI	sources/trc404_nftItem.fc	cd65c0be2d7f4ed493ab8519689f5e7ca1e0a38e
WIN	sources/init/wallet_init.fc	be9f128f957f6b1cd84396e432ffe2e26b2250f4
NII	sources/init/nftItem_init.fc	994cc717612df35a77936f8c2927690d7e1b3caf

1.3 Issue Statistic

Item	Count	Fixed	Partially Fixed	Acknowledged
Total	9	6	1	2
Informational	0	0	0	0
Minor	2	1	0	1
Medium	4	4	0	0
Major	2	0	1	1
Critical	1	1	0	0

1.4 TonBit Audit Breakdown

TonBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by NotFoundLabs to identify any potential issues and vulnerabilities in the source code of the TRC404 smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 9 issues of varying severity, listed below.

ID	Title	Severity	Status
CME-1	Incorrect Calculation of `fwd_fee`	Medium	Fixed
T4M-1	Centralization Risk	Major	Acknowledged
T4M-2	Conflict With The Jetton Standard	Minor	Fixed
T4N-1	NFT Content Error	Minor	Acknowledged
T4W-1	Jetton Balance Calculation Error	Critical	Fixed
T4W-2	No Handling of Bounced Messages	Major	Partially Fixed
T4W-3	Incorrect Gas Calculations	Medium	Fixed
T4W-4	Ignored Gas Fee	Medium	Fixed
T4N1-1	Manipulable NFT Level	Medium	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the TRC404 Smart Contract :

Admin

- The Admin can mint `Jetton` coins for other addresses that do not exceed the max supply through `mint()` .
- The Admin can change the royalty params of the collection through `change_royalty_params()` .

User

- The User can transfer their `Jetton` coins through `transfer()` .
- The User can receive their `Jetton` coins through `internal_transfer()` .
- The User can transfer their `NFT` through `transfer_nft_item()` .
- The User can change their owned `NFT` limit through `change_owned_nft_limit()` .

4 Findings

CME-1 Incorrect Calculation of `fwd_fee`

Severity: Medium

Status: Fixed

Code Location:

sources/message/common_message.fc#20

Descriptions:

In the `load_in_msg_full` function, the `fwd_fee` is not computed is loaded directly, but in the jetton standard it is computed by `muldiv(cs~load_coins(), 3, 2)`.

Suggestion:

It is recommended to refer to the jetton standard to set the `fwd_fee` value.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

T4M-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

sources/trc404_master.fc#60

Descriptions:

Centralization risk was identified in the smart contract.

- The `Admin` can mint any amount of `Jetton` tokens for other addresses, up to the maximum supply.

Suggestion:

It is recommended to take measures to mitigate this issue.

T4M-2 Conflict With The Jetton Standard

Severity: Minor

Status: Fixed

Code Location:

sources/trc404_master.fc#81;

sources/trc404_wallet.fc#298

Descriptions:

The first value in the return value of the `get_jetton_data` function in the `trc404_master` contract and the number of the return value in the `get_wallet_data` in the `trc404_wallet` contract conflict with the jetton standard. which can be confusing for users.

```
(int,int, slice, cell,cell,cell,slice,int) get_jetton_data() method_id {
    (int total_supply, slice admin_address,slice nft_collection_address, cell content, cell
jetton_wallet_code,cell nft_item_code,int mintable) = load_data();
    return (max_supply(),mintable,
admin_address,content,jetton_wallet_code,nft_item_code,nft_collection_address,total_suppl
}

(int, slice, slice, cell, cell, slice, cell, int, int, int, cell) get_wallet_data() method_id {
slice ds = get_data().begin_pars();return (ds~load_coins(),
;;jetton_balance ds~load_msg_addr(),
;;owner_address

;;pending_reduce_ jetton_balance
ds~load_dict());
;;Pending_transfer_nft_queue
}
```

Suggestion:

It is recommended to follow the jetton standard.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

T4N-1 NFT Content Error

Severity: Minor

Status: Acknowledged

Code Location:

sources/trc404_nftCollection.fc#163-165

Descriptions:

The `get_nft_content` function directly returns the incoming parameter `individual_nft_content`, should it return common content.

```
cell get_nft_content(int index, cell individual_nft_content) method_id {
    return individual_nft_content;
}
```

The implementation in the standard is as follows:

```
cell get_nft_content(int index, cell individual_nft_content) method_id {
  var (_, _, content, _, _) = load_data();
  slice cs = content.begin_parse();
  cs~load_ref();
  slice common_content = cs~load_ref().begin_parse();
  return (begin_cell()
                      .store_uint(1, 8) ;; offchain tag
                      .store_slice(common_content)
                      .store_ref(individual_nft_content)
          .end_cell());
}
```

Suggestion:

It is recommended to refer to the standard implementation of the implementation function.

Resolution:

The client replied that although this function does not add complete display logic code, it will actually perform specialized processing to display specific content based on the characteristics data of the collection contract and NFTs.

T4W-1 Jetton Balance Calculation Error

Severity: Critical

Status: Fixed

Code Location:

sources/trc404_wallet.fc#115

Descriptions:

In the `updateOwnedNftNumberAndJettonBalance` function, when the user's `Jetton_balance` is less than the number of FT's corresponding to an NFT, the protocol takes the step of zeroing out the user's `Jetton_balance` and adding the number of FT's corresponding to an NFT to the `Pending_reduce_jetton_balance` variable. However, this will actually result in subtracting more of the user's own `Jetton_balance` for the user.

```
() updateOwnedNftNumberAndJettonBalance(int msg_value, int
nft_represent_ft_amount, slice to_address, int query_id, int item_index, slice
response_address) impure inline_ref {
    Storage::Owned_nft_number = Storage::Owned_nft_number - 1;
    ;;try to reduce jetton_balance, if result is negative , add Pending_reduce_jetton_balance
by nft_represent_ft_amount
    int result = Storage::Jetton_balance - nft_represent_ft_amount;
    if (result < 0) {
        Storage::Pending_reduce_jetton_balance = Storage::Pending_reduce_jetton_balance +
nft_represent_ft_amount;
        Storage::Jetton_balance = 0;
    } else {
```

Suggestion:

It is recommended that the user's `Jetton_balance` and `Pending_reduce_jetton_balance` be properly recorded.

Resolution:

This issue has been resolved with the customer modifying the code as follows:

```
if (result < 0) {
        Storage::Pending_reduce_jetton_balance = Storage::Pending_reduce_jetton_balance +
abs(result);
        Storage::Jetton_balance = 0;
```

T4W-2 No Handling of Bounced Messages

Severity: Major

Status: Partially Fixed

Code Location:

sources/trc404_wallet.fc

Descriptions:

There are some messages where `bounce` is set to true, but these messages are not handled in the contract, which could lead to a security risk in the smart contract. Such as the message of the `transfer` function in the `trc404_wallet` contract, When user A transfers `Jetton` coins to user B, since bounce messages are not handled in the contract, if user B fails to accept the transfer, it may result in a decrease in user A's balance and no change in user B's balance, it will result in the loss of the `Jetton` tokens.

Suggestion:

It is recommended to take measures for these bounce messages as same as the Jetton standard: `https://github.com/ton-blockchain/token-contract/blob/main/ft/jetton-wallet.fc#L197`.

```
() on_bounce (slice in_msg_body) impure {
  in_msg_body~skip_bits(32); ;; 0xFFFFFFFF
  (int balance, slice owner_address, slice jetton_master_address, cell jetton_wallet_code) =
load_data();
  int op = in_msg_body~load_uint(32);
  throw_unless(709, (op == op::internal_transfer()) | (op == op::burn_notification()));
  int query_id = in_msg_body~load_uint(64);
  int jetton_amount = in_msg_body~load_coins();
  balance += jetton_amount;
  save_data(balance, owner_address, jetton_master_address, jetton_wallet_code);
}
```

Resolution:

The case of sending FTs is handled in the bounce issue, but due to TON's limited size of bounced messages, NFT-related bounced messages can not be handled.

T4W-3 Incorrect Gas Calculations

Severity: Medium

Status: Fixed

Code Location:

sources/trc404_wallet.fc#174

Descriptions:

The `trac404_wallet` contract multiplies `forward_ton_amount` with the `need_to_burn_nft_number` variable as well when calculating the `total_burn_nft_gas` variable, but `forward_ton_amount` is not part of the destruction of nft gas consumption.

```
int total_burn_nft_gas = (reduce_nft_supply_fee() + forward_ton_amount) *
(need_to_burn_nft_number);
        msg_value = msg_value - total_burn_nft_gas;
```

Suggestion:

It is recommended that `forward_ton_amount` be taken out of the formula.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

T4W-4 Ignored Gas Fee

Severity: Medium

Status: Fixed

Code Location:

sources/trc404_wallet.fc#94

Descriptions:

The `core_internal_transfer` function in the trc 404 wallet handles `forward_ton_amount` without considering `fwd_fee`.

```
if (forward_ton_amount > 0) {
        msg_value -= forward_ton_amount;
        sendMsg(bounce::false(), sendMode::PAY_GAS_SEPARATELY(),
Storage::Owner_address, forward_ton_amount, transferNotificationMsg(query_id,
jetton_amount, from_address, forward_payload));
    }
```

This is the case in the standard implementation:

```
if(forward_ton_amount) {
    msg_value -= (forward_ton_amount + fwd_fee);
    slice either_forward_payload = in_msg_body;
```

The `min_storageFee` is inconsistent with the reference implementation here.

```
int min_tons_for_storage() asm "10000000 PUSHINT"; ;; 0.01 TON
```

And the gas variables in the contract are all self-defined to calculate whether there is any error on the gas.

```
int min_storageFee()                    asm "1000000 PUSHINT";
int basice_transfer_ft_gas_consumption() asm "18000000 PUSHINT";
int transfer_one_ft_gas()               asm "5000000 PUSHINT";
int reduce_nft_supply_fee()             asm "38000000 PUSHINT";
int transfer_nft_burn_nft_fee()         asm "38000000 PUSHINT";
int add_one_ft_and_nft_gas()            asm "17600000 PUSHINT";
int receive_ft_gas_consumption()        asm "2000000 PUSHINT";
int need_add_nft_supply_action_fee()     asm "15300000 PUSHINT";
int no_need_add_nft_supply_action_fee()  asm "14800000 PUSHINT";
```

Suggestion:

It is recommended to implement it according to the standard and to consider the issue of gas calculation errors.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

T4N1-1 Manipulable NFT Level

Severity: Medium

Status: Fixed

Code Location:

sources/trc404_nftItem.fc#72

Descriptions:

In the `deployNftItem` function, the `random()` generates a pseudo-random number that can be predicted, leading to manipulation of the `NFT` level by users.

Suggestion:

It is recommended to add the `randomize_lt()` call before generating random numbers to make it unpredictable.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.