BeaverLand Audit Report

Mon Oct 14 2024

✉ contact@bitslab.xyz https://twitter.com/tonbit_

TonBit

BeaverLand Audit Report

1 Executive Summary

1.1 Project Information

Description	An NFT and Founding Vault protocol
Type	DeFi
Auditors	TonBit
Timeline	Mon Jul 15 2024 - Mon Oct 14 2024
Languages	FunC
Platform	Ton
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/TheOpenForest/tof-backend-monorepo
Commits	a687b3ff769c7cfc95a85cf4bf9802bae324717b

1.2 Files in Scope

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

ID	File	SHA-1 Hash
FVA	FundingVault/funding-vault.fc	7a2c1f3d1c58f5960fb2f9c65a041e85a0e48a72
NIT	NftMinter/contracts/nft_item.fc	6f6b61d91bac47f03f87daf38712a7e63a59ffe1
SIT	NftMinter/contracts/sbt_item.fc	5f4083631a65a9d83f99d5706b5eb203bbcdfaae
PAR	NftMinter/contracts/imports/params.fc	cdbd4dd247d21de4de6ef4bc40bf2e97f5e723dd
OCO	NftMinter/contracts/imports/op_codes.fc	bba6c3d26fcce83e956d272d281dc4d4cd36fffe
STD	NftMinter/contracts/imports/stdlib.fc	aec062c0934591fa31de86353f23144dc4c3039d
NMI	NftMinter/contracts/nft_minter.fc	218534058c45d65efae5ea4536146cf327ff33b4

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	2	0	2
Informational	1	0	1
Minor	0	0	0
Medium	1	0	1
Major	0	0	0
Critical	0	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 BeaverLand to identify any potential issues and vulnerabilities in the source code of the Tof-Backend-Monorepo 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 2 issues of varying severity, listed below.

ID	Title	Severity	Status
FVA-1	Batch Transaction Execution Implementation	Medium	Acknowledged
FVA-2	The Code Is Not Fully Implemented	Informational	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the Tof-Backend-Monorepo Smart Contract :

Owner

- The owner can change the public key through an `op == 86` message.
- The owner can deploy a new NFT through an `op == 1` message.
- The owner can batch deploy new NFTs through an `op == 2` message.
- The owner can change the owner through an `op == 3` message.

User

- The user can withdraw money with a signature through an internal message from the funding vault.
- The user can send a transaction message through an external message from the funding vault.
- The user can transfer NFT ownership through an `op::transfer` message.
- The user can deploy a new NFT with a signature through an `op == 87` message.
- The user can request ownership information through an `op::request_owner` message.
- The user can prove ownership through an `op::prove_ownership` message.
- The user can get static data through an `op::get_static_data` message.
- The user can destroy the NFT through an `op::destroy` message.
- The user can revoke the NFT through an `op::revoke` message.
- The user can take excess funds through an `op::take_excess` message.

4 Findings

FVA-1 Batch Transaction Execution Implementation

Severity: Medium

Status: Acknowledged

Code Location:

FundingVault/funding-vault.fc#55-90

Descriptions:

The external message in the FoundingVault contract is designed to allow users to execute multiple transactions simultaneously, similar to Highload-wallet v2: Highload-wallet v2 Code.

```
int i = -1;
do {
  (i, var cs, var f) = dict.idict_get_next?(16, i);
  if (f) {
    var mode = cs~load_uint(8);
    send_raw_message(cs~load_ref(), mode);
  }
} until (~ f);
```

Consider using the Highload-wallet code implementation for other parts of the current contract as well.

Suggestion:

It is recommed to consider referring to the implementation approach of High-Load Wallets. Another key point to note is that,

> Higload-v2 deprecation.
> Since most of services already migrated to highload v3 wallet,
> old (unsafe) highload-wallet-v2 is finally deprecated.
> It is planned to conduct one-time unlock for wallets that are stuck
> due to too many requests per second through a time-limited gas
> limit increase.

FVA-2 The Code Is Not Fully Implemented

Severity: Informational

Status: Acknowledged

Code Location:

FundingVault/funding-vault.fc#3-54

Descriptions:

The internal message in the FundingValul contract is not fully implemented yet.

```
;; TODO: 1. Check if the nonce is valid
;;       2. Set nonce to invalid
;;       3. Check if it is a jetton
;;       4. If it is a jetton, send a transaction to the jetton_wallet_address
;;       5. If it is not a jetton, send a transaction to the sender_address
```

Suggestion:

It is recommended to implement all the code

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.