Nirvana Smart Contract

Audit Report

Fri Sep 27 2024

contact@bitslab.xyz https://twitter.com/scalebit_

ScaleBit

Nirvana Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	
Type	AMM
Auditors	ScaleBit
Timeline	Wed Sep 18 2024 - Wed Sep 18 2024
Languages	Rust
Platform	Solana
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/nirvanadao/n-vana/
Commits	d21f215d9f9e160d6caccc63b8c087414e99b42e

1.2 Files in Scope

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

ID	File	SHA-1 Hash
LIB	sol/programs/nvana/src/lib.rs	48d20eb7660566b2e2cc5cf8296a6da3283acf51
UTI	sol/programs/nvana/src/util.rs	5c16ae24ad33566352f31c1a502a5aa2afcb4c16
PAC	sol/programs/nvana/src/state/personal_account.rs	84495556e271ee4562721dfa8ad449a838f1e8a7
ARE	sol/programs/nvana/src/state/alms_rewarder.rs	c046d16609a87ba316661fa81d0dde8fa49cf9bb
MRE	sol/programs/nvana/src/state/metta_rewarder.rs	ee4ce69ae8487e9e2bab13a42324e08abf09c957
MOD	sol/programs/nvana/src/state/mod.rs	0c782a0f6a305dbd434a78ef17ca96a6008831fa
NUM	sol/programs/nvana/src/state/number.rs	caa3685cde989fa6dec4bb5621ce1a221c46e762
COM	sol/programs/nvana/src/state/common.rs	64b053839514324da10116910f542ebf137763f9
TEN	sol/programs/nvana/src/state/tenant.rs	62f0670434dfb746b94ba6c557d74300cfa80088
EPR	sol/programs/nvana/src/instructions/market/execute_prana.rs	d0c74c9d96ccc1dd210cbcdc257ece12c1d4d486
RFL	sol/programs/nvana/src/instructions/market/raise_floor.rs	ed126898af074bf1af440727eb72ddc49566cae5

BUY	sol/programs/nvana/src/instructions/market/buy.rs	06b03d771847d0658517ce8ccc1a161a57a63c91
SEL	sol/programs/nvana/src/instructions/market/sell.rs	b5bada91bd67a94ef92f53846cfc8aab72aa6128
SPNSIMMR	sol/programs/nvana/src/instructions/market/mod.rs	f3c7da0a6830428531e672a7b8e94953acd43f1a
ASP	sol/programs/nvana/src/instructions/admin/admin_set_params.rs	323db674c665c8294f86b6cc9392f1d0bf747304
ASC	sol/programs/nvana/src/instructions/admin/admin_set_controls.rs	07c49e9a7dca7af8d6e82a605b3d0dbb2ef066cb
SPNSIAMR	sol/programs/nvana/src/instructions/admin/mod.rs	4439968de310195622deed8859a141a74827e0d0
MPR	sol/programs/nvana/src/instructions/admin/mint_prana.rs	f22dcd1a58fda175f37507d2f6dd9a30ab0e8c88
ITE	sol/programs/nvana/src/instructions/init_tenant.rs	65f448663ee84637ca8b93d727f05474b47c011c
CRP	sol/programs/nvana/src/instructions/personal_account/collect_rev_prana.rs	0bcf42172536ceeab179d28f122a8b78d01130a7
WAN	sol/programs/nvana/src/instructions/personal_account/withdraw_ana.rs	cd8170d151e6f3453e363a2ad756eb710f8beff5
WPR	sol/programs/nvana/src/instructions/personal_account/withdraw_prana.rs	38259b32b4fcf97c8a2011b74c1ff24ef24b985b
DPR	sol/programs/nvana/src/instructions/personal_account/deposit_prana.rs	fe5a871b266f11ac7151c59e026b0c96202b916c

CPR	sol/programs/nvana/src/instructions/personal_account/collect_prana.rs	2d656909ea14f9be804e8c9bd7179bf898bf40cd
SPNSIPAMR	sol/programs/nvana/src/instructions/personal_account/mod.rs	2f7098578db3e0e2750487f87105f5df8043eb1b
DAN	sol/programs/nvana/src/instructions/personal_account/deposit_ana.rs	7360205180e5d255432b70c486f694e622855acc
IPA	sol/programs/nvana/src/instructions/personal_account/init_personal_account.rs	25b605b62ec393fa776b19dd1c6eaa3a7e3b1b74
SPR	sol/programs/nvana/src/instructions/personal_account/stage_prana.rs	e0c28f9bd943ab8fe524a5cba84c20e2493f5b41
SVO	sol/programs/nvana/src/instructions/personal_account/set_votes.rs	6a592045a26386bdc42cdfdb9d5c60b69566ea5b
SRP	sol/programs/nvana/src/instructions/personal_account/stage_rev_prana.rs	5d9a313ca6838d3b3de88474f04705e15d124431
TVO	sol/programs/nvana/src/instructions/tally_votes.rs	4d236773ef10b10e3bdde4c9a64bc6710e1d32f1
RNI	sol/programs/nvana/src/instructions/nirv/repay_nirv.rs	e8e6b0191d1fee331ef6290c8498664c69c2fb74
SPNSINMR	sol/programs/nvana/src/instructions/nirv/mod.rs	aaa8dd8e45f6e3e1e5a9a7340f8579243f49be75
BNI	sol/programs/nvana/src/instructions/nirv/borrow_nirv.rs	cc25d0a7c1a5f37730653d4527b4795fbc04f7c3
SPNSIMR	sol/programs/nvana/src/instructio	4f6ef332d21891341c63ac50a9808

	ns/mod.rs	9e59ae6546d
WME	sol/programs/nvana/src/instructions/metta/withdraw_metta.rs	ada5e6fd9c154b0354ce566bbce7e8ae8383dffc
DME	sol/programs/nvana/src/instructions/metta/deposit_metta.rs	eed7a0e881d6bdf858878094e9547cd2459455ba
SPNSIMMR	sol/programs/nvana/src/instructions/metta/mod.rs	5580ca6a99c595897c1953588a6cc56a5a7c7b5d
SMR	sol/programs/nvana/src/instructions/metta/stage_metta_rev.rs	8f9e9a1df5f5b4b3a3655ca15e973307d36882eb
IMR	sol/programs/nvana/src/instructions/metta/init_metta_rewarder.rs	b8b2a336b8e7924b1afaee2df223eea6de9210f2
CMR	sol/programs/nvana/src/instructions/metta/collect_metta_rev.rs	2b0d0cf1880fe8e11ca18c5edb65c0b63947d392
OUR	sol/programs/nvana/src/instructions/ouroboros.rs	9c096d14756fa1a452411adaa222c556d6ee1456
SPNSIACARR	sol/programs/nvana/src/instructions/alms/collect_alms_rev.rs	9021451c62dfe421d03e9c665a2746c77ffc5e71
DAL	sol/programs/nvana/src/instructions/alms/deposit_alms.rs	618441cff50c8c47fa5f30fc0873f34a352c96e5
SAR	sol/programs/nvana/src/instructions/alms/stage_alms_rev.rs	e9839d33c24f718b862f20d333d1b62fda98f448
IAR	sol/programs/nvana/src/instructions/alms/init_alms_rewarder.rs	4e0fd982d146d11871515e20cfd2b7d0c4c50c7d
SPNSIAMR	sol/programs/nvana/src/instructions/alms/mod.rs	501ab642250ea12318e42aae92ebd51c7ce69448
WAL	sol/programs/nvana/src/instructions/alms/withdraw_alms.rs	081202d19afad4d46b109054b1a95a17043ca4d0

PDA	sol/programs/nvana/src/pda.rs	6bad25d1207e2c91aa2217f44705bbad5541a252

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	4	3	1
Informational	0	0	0
Minor	3	2	1
Medium	1	1	0
Major	0	0	0
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit 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
- Number of rounding errors
- Functionality Checks
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Replay attacks
- Coding style issues

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 Nirvana to identify any potential issues and vulnerabilities in the source code of the Nirvana Smart Contract 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 4 issues of varying severity, listed below.

ID	Title	Severity	Status
ASP-1	Centralization Risk	Minor	Acknowledged
SVO-1	Unable to Vote on Floor Price	Medium	Fixed
SVO-2	The Actual Voting Weight Is Less Than The Calculated Voting Weight	Minor	Fixed
TEN-1	Unused Fields	Minor	Fixed

3 Participant Process

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

Admin

- The `Admin` can initialize a tenant through `init_tenant()`.
- The `Admin` can set the contract parameters through `admin_set_params()`.
- The `Admin` can manage contract lending and collateralization and purchase permissions through `admin_set_controls()`.
- The `Admin` can mint prANA through `mint_prana()`.

The User

- The `User` can buy ANA through `buy()`.
- The `User` can sell owned ANA through `sell()`.
- The `User` can initialize their owned personal account through `init_personal_account()`.
- The `User` can deposit ANA through `deposit_ana()`.
- The `User` can withdraw ANA through `withdraw_ana()`.
- The `User` can borrow nirv through `borrow_nirv()`.
- The `User` can repay nirv through `repay_nirv()`.
- The `User` can update the reward when ANA deposited through `stage_prana()`.
- The `User` can collect rewards through `collect_prana()`.
- The `User` can use prANA to buy ANA at floor price through `execute_prana()`.
- The `User` can initialize the reward manager through `init_alms_rewarder()`.
- The `User` can collect rewards through `stage_alms_rev()`.
- The `User` can deposit alms through `deposit_alms()`.
- The `User` can withdraw alms through `withdraw_alms()`.
- The `User` can collect rewards through `collect_alms_rev()`.

- The `User` can deposit prANA to vote for the fee parameters through `set_votes()`.
- The `User` can execution of voting results through `tally_votes()`.
- The `User` can deposit prANA through `deposit_prana()`.
- The `User` can withdraw prANA through `withdraw_prana()`.
- The `User` can raise the floor price if it meet the conditions through `raise_floor()`.
- The `User` can deposit metta through `deposit_metta()`.
- The `User` can withdraw metta through `withdraw_metta()`.
- The `User` can collect rewards through `collect_metta_rev()`.
- The `User` can initialize the reward manager through `init_metta_rewarder()`.
- The `User` can update the revenue for prANA on a personal account through `stage_rev_prana()`.
- The `User` can collect the revenue for prana on a personal account through `collect_rev_prana()`.
- The `User` can increase ramp start when reserve fund surplus through `ouroboros()`.

4 Findings

ASP-1 Centralization Risk

Severity: Minor

Status: Acknowledged

Code Location:

sol/programs/nvana/src/instructions/admin/admin_set_params.rs;

sol/programs/nvana/src/instructions/admin/admin_set_controls.rs;

sol/programs/nvana/src/instructions/admin/mint_prana.rs

Descriptions:

Centralization risk was identified in the smart contract.

- Admin can suspend ANA's `buy` and `sell` functions, as well as nirv's `borrow` and `realize` functions.
- Admin can mint any number of prANA tokens to any address.

Suggestion:

It is recommended to take ways to reduce the risk of centralization.

SVO-1 Unable to Vote on Floor Price

Severity: Medium

Status: Fixed

Code Location:

sol/programs/nvana/src/instructions/personal_account/set_votes.rs;

sol/programs/nvana/src/instructions/init_tenant.rs

Descriptions:

There is no function that can change the values of fields `floor_raise_yes` and `floor_raise_no` in strcut `GlobalBallot` and cannot initialize properly, which involves modifying the floor price.This may cause the floor price modification to not work properly.

Suggestion:

It is recommended to ensure this design meets your requirements.

Resolution:

The customer accepted our suggestion and fixed this issue in a subsequent commit.

SVO-2 The Actual Voting Weight Is Less Than The Calculated Voting Weight

Severity: Minor

Status: Fixed

Code Location:

sol/programs/nvana/src/instructions/personal_account/set_votes.rs#40

Descriptions:

When `prANA` stakers vote to change the fee parameters, the `alter_votes` function is called.

```
op(&mut self.buy_ana_fee_mbps, &votes.buy_ana_fee_mbps);
op(&mut self.sell_ana_fee_mbps, &votes.sell_ana_fee_mbps);
```

It only changes two fields.This raises two problems

1. The only fields that can actually be voted on are `buy_ana_fee_mbps` and `sell_ana_fee_mbps` .
2. When calculating whether the user's voting power exceeds the staked 'prANA', it is not true because only two fields have changed, and the other fields have not changed.
3. There are no other functions that can change other fields.

Suggestion:

It is recommended to compare the effective vote count with the stake count, and ensure that other fields make sense and can be modified.

Resolution:

The customer fixed this problem in subsequent development.

TEN-1 Unused Fields

Severity: Minor

Status: Fixed

Code Location:

sol/programs/nvana/src/state/tenant.rs#59

Descriptions:

In the Tenant structure, there are two fields with the same name, namely:

1. `tenant.prana_apr_mbps`
2. `tenant.gov.ballot.prana_apr_mbps`

When calculating the apy of prANA, the value of the first parameter is used

```
let partial_apr = Number::from_partial_apr(self.prana_apr_mbps.into(), delta);
```

The second value is not used anywhere else, it is an unused field.

Suggestion:

It is recommended to ensures that this complies with your design and considers further optimizing the code.

Resolution:

The customer accepted our suggestion and fixed this issue in a subsequent commit.

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.