MOVEBIT
Securing the Move Ecosystem

SuiPad Smart Contract **Audit Report**

https://twitter.com/movebit_

contact@movebit.xyz

SuiPad Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	A launchpad project on Sui.
Type	Launchpad
Auditors	MoveBit
Timeline	Apr 19, 2023 – May 11, 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	Repository: https://github.com/SuiPad/suipad-contract Received Commit: 6e7615242264cacb733687e5d995a3b7039b1841 Last Reviewed Commit: 6e86ecd684977ea922d5be3320b9efd5fd0e19f7

1.2 Files in Scope

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

ID	Files	SHA-1 Hash
CPG	sources/campaign.move	da97e204fe5d4600ddbd067c917ab9c5d44e2a91
LPD	sources/launchpad.move	4438b9bdebedded8e5e02ae381ede91182811bb9
ORA	sources/oracle.move	59a8d97dc2c5f9b3dc4af94cb063cc390dc7bcbd
TOK	sources/token.move	1b5370a1d8f99d11256d35f4cf54ffc63f6c6618
WHT	sources/whitelist.move	aa97da99167e73d1ba3a3cf441f13b309ede5161
ISR	Launchpad-nostake/sources/insurance.move	b7ed47a8d88ec1084b47fd8e8e2f8c61a321cb7e
VAT	Launchpad-nostake/sources/vault.move	a75c18f88e42782077724226 3dcf9b0134067f49
STK	Launchpad/sources/staking.move	e725a4e98b17f61117f8be2cf0 0766d7ea11c617
CPG_STAKING	Launchpad/sources/campaign.move	37bf04f9ca64177fa336fc758 84a9f1abbfb6887
VAT_STAKING	Launchpad/sources/vault.move	b5ea69a2ef9228d58e271f172b26da5f31b013ec
WHT_STAKING	Launchpad/sources/whitelist.move	17f22c74abe9a58ffd71f72830aa968307aa9f28
LPD_STAKING	Launchpad/sources/launchpad.move	8ef52d7ba6d3b97abce1a713a2b29dcd8082ca29

ISR_STAKING	Launchpad/sources/insurance.move	3c5908010fc91befe50cdfd1885f6906168982dd
SUP	sources/suip.move	e030b35fea981292a9379512675d3726257afc32

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	26	21	5
Informational	3	2	1
Minor	7	6	1
Medium	2	1	1
Major	9	7	2
Critical	5	5	

1.4 MoveBit Audit BreakDown

MoveBit 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
- The flow of capability
- Witness Type

1.5 Methodology

The security team adopted the "**Testing and Automated Analysis**", "**Code Review**" and "**Formal Verification**" 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 sees in section **1.2**.

(3) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) 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 **SuiPad** to identify any potential issues and vulnerabilities in the source code of the **SuiPad LaunchPad** 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 have identified 26 issues of varying severity, listed below.

ID	Title	Severity	Status
CPG-01	Lack of Validation for `target_amount` and `tokens_to_sell` in `create_campaign`	Critical	Fixed
ISR-02	Function Can't be Called	Critical	Fixed
ISR-03	Lack of State Changes During Function Execution	Critical	Fixed
VAT-04	Incorrect Formula	Critical	Fixed
LPD_STAKING-05	Missing Test Comments	Critical	Fixed
CPG-06	Incorrect Function Visibility	Major	Fixed
CPG-07	Accuracy Loss	Major	Fixed
CPG-08	Incorrect Calculation of `amount_to_claim` in `claim`	Major	Fixed

CPG-09	Lack of Validation for Campaign and Whitelist ID in `invest`	Major	Fixed
CPG-10	Lack of Validation for Funding Status in `fund`	Major	Fixed
CPG_STAKING-11	Incorrect Variable Assignment	Major	Fixed
STK-12	Incorrect Function Visibility	Major	Fixed
CPG-13	Missing Permissions Verification in `fund` Function	Medium	Acknowledged
CPG-14	Lack of Validation for Campaign Status in `invest`	Medium	Fixed
WHT-15	Lack of Validation for Existing Whitelist Member	Minor	Fixed
CPG-16	Unchecked Vector	Minor	Fixed
CPG-17	Precision Loss	Minor	Fixed
TOK-18	Incorrect Parameter Configuration	Minor	Fixed
CPG_STAKING-19	Unused Struct	Minor	Acknowledged
STK-20	Accuracy Loss	Minor	Fixed
CPG-21	Redundant Code in `else` Branch in `claim_rewards`	Informational	Fixed
WHT-22	Missing Emit Events	Informational	Acknowledged

CPG-23	Unused Variables	Informational	Fixed
ISR-24	Inability to Retrieve Unused Tokens in `fund.vault` and `Vault.reward_balance`	Major	Acknowledged
ISR-25	Lack of Parameter Check	Major	Acknowledged
CPG_STAKING-26	Lack of Event	Minor	Fixed

3 Participant Process

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

Admin

- Admin can create a new campaign through `create_campaign<TI, TR>()` .
- Admin can fund a campaign through `fund<TI, TR>()` .
- Admin can add investors to the whitelist through `add_bulk_to_whitelist<TI, TR>()` and `add_to_whitelist<TI, TR>()` .
- Admin can set the allocations for the campaign through `set_allocations<TI, TR>()` .
- Admin can issue refund allowance through `issue_refund_allowance()` .
- Admin can add coins into the fund vault through `add_coins<T>` .
- Admin can set the `penalty` receiver through `set_penalty_receiver()` .

Investor

- Investor can create a new stake through `new_stake()` .
- Investor can extend the stake through `extend_stake()` .
- Investor can withdraw the stake through `withdraw()` .
- Investor can apply for the whitelist through `apply_for_whitelist<TI, TR>()` .
- Investor can invest in a campaign through `invest<TI, TR>()` and `second_sale_invest<TI, TR>()` .

- Investor can claim the reward from the campaign through `claim_rewards<TI, TR>()`.
- Investor can claim insurance if insured through `claim_insurance<TI, TR>()`.

Receiver

- Receiver can claim investment from the campaign through `claim_investment<TI, TR>()`.

4 Findings

CPG-01 Lack of Validation for `target_amount` and `tokens_to_sell` in `create_campaign`

Severity: Critical

Status: Fixed

Code Location: sources/campaign.move#L107.

Descriptions: In the `create_campaign` method, there is a lack of validation for the `target_amount` and `tokens_to_sell` parameters, which may lead to a `target_amount * 1000 / tokens_to_sell` equaling 0 due to precision issues. This can cause users to fail to claim tokens and lock up their assets.

Suggestion: It is recommended to add a check to ensure that the result of `get_token_price` can not be zero in `create_campaign`.

Resolution: The client followed our suggestion and fixed this issue in commit `1a4d683052b58f19c6532acd39c0b2528580ece1`.

ISR-02 Function Can't be Called

Severity: Critical

Status: Fixed

Code Location: Launchpad-nostake/sources/insurance.move#L99.

Descriptions: The `claim_refund` function can not be called because The parameter `Vault<TI, TR>` cannot be passed into the function, resulting in coins deposited into the fund that will not be retrieved.
Suggestion: It is recommended that modify the design of the function.

Resolution: The client followed our suggestion and fixed this issue in commit `1a4d683052b58f19c6532acd39c0b2528580ece1`.

ISR-03 Lack of State Changes During Function Execution

Severity: Critical

Status: Fixed
Code Location: Launchpad-nostake/sources/insurance.move#L99.
Descriptions: In the `claim_refund` function, there is no update of the variables and no verification that the `InvestCertificate` is claimed, it will result in the user being able to use an `InvestCertificate` for multiple claims.
Suggestion: It is recommended that update the variables in the `InvestCertificate` or verify that the `InvestCertificate` is claimed.

Resolution: The client followed our suggestion and fixed this issue in commit `1a4d683052b58f19c6532acd39c0b2528580ece1`.

VAT-04 Incorrect Formula

Severity: Critical

Status: Fixed

Code Location: Launchpad-nostake/sources/vault.move#L140.

Descriptions: When refunding from `vault.investment_balance`, the `amount_to_refund` should be divided by the `DecimalPrecision` after being multiplied by it, it will result in the user acquiring a larger number of refunds.
Suggestion: When calculating amounts related to tokens, use the correct multiplier.

Resolution: The client followed our suggestion and fixed this issue in commit `1a4d683052b58f19c6532acd39c0b2528580ece1`.

LPD_STAKING-05 Missing Test Comments

Severity: Critical

Status: Fixed

Code Location: Launchpad/sources/launchpad.move#L23

Descriptions: The test code should be commented with #[test_only], it will result in administrator privileges that can be assigned to anyone.

Suggestion: It is recommended to add comments to the test code.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

CPG-06 Incorrect Function Visibility

Severity: Major

Status: Fixed

Code Location: sources/campaign.move#L253, sources/whitelist.move#L54.

Descriptions: The `add_to_whitelist` and `close_campaign` functions primarily use the public functions `add_investor` and `close`, however, calls to the public functions `add_investor` and `close` will result in anyone being able to whitelist additions and close the campaign.

Suggestion: It is recommended to change the visibility of public functions called within a function to internal or friend. In addition, `new` function in `campaign.move` L81 can be declared as internal if there's no further intention.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG-07 Accuracy Loss

Severity: Major

Status: Fixed

Code Location: sources/campaign.move#L205.

Descriptions: In the `claim_rewards` method, the `amount_to_claim` calculation has a precision loss, in the condition of `campaign.invested_amount > campaign.target_amount`, the value of `cert.deposit` is mostly less than `campaign.invested_amount`, making the result 0, which will prevent the user from claiming the reward.

Suggestion: It is recommended that calculations be performed using multiplication followed by division to reduce accuracy loss.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG-08 Incorrect Calculation of `amount_to_claim` in `claim`

Severity: Major

Status: Fixed
Code Location: sources/campaign.move#L200.
Descriptions: The calculation formula for `amount_to_claim` in the `claim` function is incorrect, which will result in differences in the results. Specifically, when the condition `campaign.invested_amount <= campaign.target_amount is` satisfied, `amount_to_claim` should be calculated as `(1000*cert.deposit / get_token_price(campaign))`.
Suggestion: It is recommended to modify the code and do thorough tests to ensure that the corrected formula produces accurate results.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG-09 Lack of Validation for Campaign and Whitelist ID in `invest`

Severity: Major

Status: Fixed
Code Location: sources/campaign.move#L158.
Descriptions: In the `invest` method, there is no validation that the ID of the campaign and whitelist match, which can allow members of other campaigns' whitelists to participate in the current campaign.
Suggestion: It is recommended to add a check in the `invest` function to ensure that the ID of the campaign and whitelist match before allowing the user to invest in the campaign. This will prevent members of other campaigns' whitelists from participating in the current campaign.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG-10 Lack of Validation for Funding Status in `fund`

Severity: Major

Status: Fixed
Code Location: sources/campaign.move#L137.
Descriptions: The `fund` method in the contract does not validate whether the campaign has already been fully funded, allowing multiple funding transactions. However, during distribution, the contract can only distribute a fixed amount of tokens, which may result in confusion and incorrect token distribution.
Suggestion: It is recommended to add a check in the `fund` function to ensure that the campaign has not already been fully funded before accepting any further funding. This will prevent multiple funding transactions and ensure proper token distribution during the distribution phase.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG_STAKING-11 Incorrect Variable Assignment

Severity: Major

Status: Fixed
Code Location: Launchpad/sources/campaign.move#L290.
Descriptions: In the `is_whitelist_phase` function, the value of `one_day` is 0, it should be 24*60*60*100.It will cause the result of the condition "campaign.sale_start – one_day > clock::timestamp_ms(clock)" to be incorrect.
Suggestion: It is recommended to modify the value of `one_day`.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

STK-12 Incorrect Function Visibility

Severity: Major

Status: Fixed
Code Location: Launchpad/sources/staking.move#L189.
Descriptions: The Visibility of the `update_last_distribution_timestamp` function is

public, the `last_distribution_timestamp` can be changed by the staker. It will cause skip conditional judgment in `withdraw` function, L160, so the staker will not send penalties to the penalty receiver.
Suggestion: It is recommended to modify the visibility of the `update_last_distribution_timestamp` function to internal.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

CPG-13 Missing Permissions Verification in `fund` Function

Severity: Medium

Status: Acknowledged
Code Location: sources/campaign.move#L137.
Descriptions: The `fund` method does not check for calling permissions, allowing everyone to fund the campaign.
Suggestion: It is recommended to confirm if it aligns with the design.

CPG-14 Lack of Validation for Campaign Status in `invest`

Severity: Medium

Status: Fixed
Code Location: sources/campaign.move#L158.
Descriptions: The `invest` method in the contract does not validate whether the campaign has already been closed, allowing the user to invest when the campaign has been closed, which may result in confusion and incorrect token distribution.
Suggestion: It is recommended to add a check in the `invest` method to ensure that the campaign has not already been closed before investing. This will prevent the users from investing after the campaign is closed and ensure proper token distribution during the distribution phase.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

WHT-15 Lack of Validation for Existing Whitelist Member

Severity: Minor

Status: Fixed
Code Location: sources/whitelist.move#L55.
Descriptions: In the method `add_investor`, there is no check for the existence of the added address. This may lead to adding the same address repeatedly.
Suggestion: It is recommended to add a check for duplicate addresses to avoid adding the same address multiple times.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

CPG-16 Unchecked Vector

Severity: Minor

Status: Fixed
Code Location: Launchpad-nostake/sources/campaign.move#L84.
Descriptions: When creating a campaign, did not check if `scheduled_times` and `scheduled_rewards` meet the following requirements, the vector lengths are equal, the sum of `scheduled_rewards` is 100, and `scheduled_times` increments. In `vault.move` L108, when `scheduled_times` is longer than the length of `Bscheduled_rewards`, the array `vault.scheduled_rewards` will report an error in the loop.
Suggestion: It is recommended to check if the vector meets the above conditions before creating a campaign.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

CPG-17 Precision Loss

Severity: Minor

Status: Fixed
Code Location: Launchpad-nostake/sources/campaign.move#L191; Launchpad-nostake/sources/vault.move#L112.
Descriptions: Divide first and then multiply will lose precision in the operation.

Suggestion: It is recommended that calculations be performed using multiplication followed by division to reduce accuracy loss.

Resolution: The client followed our suggestion and fixed this issue in commit `1a4d683052b58f19c6532acd39c0b2528580ece1`.

TOK-18 Incorrect Parameter Configuration

Severity: Minor

Status: Fixed
Code Location: sources/token.move#L15.
Descriptions: The description of the token is incorrect, It should not be `SuiPad launchpad test token`, but `SuiPad launchpad token`.
Suggestion: It is recommended that modify the description of the token.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

CPG_STAKING-19 Unused Struct

Severity: Minor

Status: Acknowledged
Code Location: Launchpad/sources/campaign.move#L44, L58, 65; Launchpad-nostake/sources/campaign.move#L40, L54, L61 ; Launchpad/sources/vault.move#L35; Launchpad-nostake/sources/vault.move#L35.
Descriptions: The struct `CampaignClosedEvent`, `RewardsClaimedEvent`, `InvestmentClaimedEvent` are unused in `campaign.move`, `RefundInvestmentEvent` in `vault.move`.
Suggestion: It is recommended to delete unused struct.

STK-20 Accuracy Loss

Severity: Minor

Status: Fixed
Code Location: Launchpad/sources/staking.move#L162.
Descriptions: In the `withdraw` function, there is an accuracy loss in the calculation of `lock.amount / 100 * pool.investment_lock_penalty`, this calculation should be multiplied first

and then divided.
Suggestion: It is recommended that calculations be performed using multiplication followed by division to reduce accuracy loss.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

CPG-21 Redundant Code in `else` Branch in `claim_rewards`

Severity: Informational

Status: **Fixed**
Code Location: sources/campaign.move#L204.

Descriptions: In the `claim_rewards` method, `else if (campaign.invested_amount > campaign.target_amount)` has the same effect as `else` here, using `else` instead of `else if` can improve program efficiency.
Suggestion: It is recommended to use `else` instead of `else if (campaign.invested_amount > campaign.target_amount)`.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

WHT-22 Missing Emit Events

Severity: Informational

Status: **Acknowledged**
Code Location: sources/whitelist.move#L45.
Descriptions: The smart contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track important actions or detect potential issues.
For example:

- `add_to_whitelist` in the `whitelist.move`

Suggestion: It is recommended to emit an event for this sensitive action.

WHT-23 Unused Variables

Severity: Informational

Status: Fixed

Code Location: Launchpad-nostake/sources/campaign.move#L19; Launchpad-nostake/sources/insurance.move#L16; Launchpad-nostake/sources/launchpad.move#L6.

Descriptions: There are many unused variables in the file, such as: in the campaign.move, L19 `EOnlyReceiver`, L24 `ENotEnoughFunds`, L25 `EInvestmentAlreadyClaimed`, L26 `EAlreadyRequested`, L28 `EAllocationExceed`, L29 `EInvalidAllocationsLength`, in the insurance.move, L16 `DecimalPrecision`, in the launchpad.move, L6 `ENotCaimpaignAdmin`.

Suggestion: It is recommended that remove unused variables.

Resolution: The client followed our suggestion and fixed this issue in commit `e8217582c2b8bbd6777cd7cf49bf21beca21f119`.

ISR-24 Inability to Retrieve Unused Tokens in `fund.vault` and `Vault.reward_balance`

Severity: Major

Status: Acknowledged

Code Location: Launchpad-nostake/sources/campaign.move, Launchpad/sources/insurance.move.

Descriptions: The current implementation lacks functionality to retrieve unused tokens in `fund.vault` and `Vault.reward_balance`. This means that any tokens remaining in these accounts cannot be accessed or reclaimed by the users.

Suggestion: It is recommended to implement a mechanism that allows the retrieval of unused tokens in both `fund.vault` and `Vault.reward_balance`.

ISR-25 Lack of Parameter Check

Severity: Major

Status: Acknowledged

Code Location: Launchpad/sources/staking.move#L165-L168.

Descriptions: In the withdraw function, the result of the `penalty` calculation may be greater than `lock.amount`, resulting in the inability to withdraw the stake coins.

Suggestion: It is recommended to check the result of the `penalty` calculation.

CPG_STAKING-26 Lack of Event

Severity: Minor

Status: Fixed

Code Location: Launchpad/sources/campaign.move#L187.

Descriptions: In the `set_allocations` function, there is a lack of an event.

Suggestion: It is recommended to add an event for the function.

Resolution: The client followed our suggestion and fixed this issue in commit `6e86ecd684977ea922d5be3320b9efd5fd0e19f7`.

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.
- **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.

MOVEBIT

Securing the Move Ecosystem

https://twitter.com/movebit_

contact@movebit.xyz