smart state

Web3 security easier than ever

Near Genesis
Smart contract audit report

May 18 2023

Table of contents

Methodology

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- FA2 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unused vars
- Unchecked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions/Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

Structure of contract

internal.rs

Contract methods analysis:

```
internal_ft_transfer(
  &mut self,
  project_id: u64,
  account_id: &AccountId,
  token_account_id:
  &AccountId,
  amount: Balance,
  is_in_token: bool
 )
```

Vulnerabilities not detected

```
after_ft_transfer(
  &mut self,
  project_id: u64,
  account_id: AccountId,
  token_account_id:
  AccountId,
  amount: WrappedBalance,
  is_in_token: bool
 )
```

Vulnerabilities not detected

```
after_is_approved(
  &mut self,
  #[callback] is_approved:
  bool,
  account_id: AccountId,
  attached_deposit:
  WrappedBalance
 )
```

Vulnerabilities not detected

```
ft_on_transfer(
  &mut self,
  sender_id: ValidAccountId,
  amount: U128,
  msg: String
 )
```

Vulnerabilities not detected

Structure of contract

lib.rs

Contract methods analysis:

new(astro_id: ValidAccountId, listing_fee: WrappedBalance, astrodao_fee: WrappedBalance)
Vulnerabilities not detected

assert_owner(&self)
Vulnerabilities not detected

assert_astro(&self)
Vulnerabilities not detected

assert_admin(&self)
Vulnerabilities not detected

append_admin_account_id(&mut self, admin_account_id: AccountId)
Vulnerabilities not detected

remove_admin_account_id(&mut self, admin_account_id: AccountId)
Vulnerabilities not detected

get_admin_account_ids(&self)
Vulnerabilities not detected

append_token_account_id(&mut self, token_account_id: AccountId)
Vulnerabilities not detected

remove_token_account_id(&mut self, token_account_id: AccountId)
Vulnerabilities not detected

get_token_account_ids(&self)
Vulnerabilities not detected

`set_astrodao_account(&mut self, astro_id: ValidAccountId)`
Vulnerabilities not detected

`get_astrodao_account(&self)`
Vulnerabilities not detected

`set_astrodao_fee(&mut self, astrodao_fee: WrappedBalance)`
Vulnerabilities not detected

`get_astrodao_fee(&self)`
Vulnerabilities not detected

`set_listing_fee(&mut self, listing_fee: WrappedBalance)`
Vulnerabilities not detected

`get_listing_fee(&self)`
Vulnerabilities not detected

Structure of contract

project.rs

Contract methods analysis:

assert_valid_not_started(&self)
Vulnerabilities not detected

from_input(project_id: u64, project_input: ProjectInput, owner_id: AccountId)
Vulnerabilities not detected

into_output(self, project_id: u64)
Vulnerabilities not detected

is_started(&self)
Vulnerabilities not detected

is_ended(&self)
Vulnerabilities not detected

internal_unwrap_project(&self, project_id: u64)
Vulnerabilities not detected

internal_get_project(&self, project_id: u64)
Vulnerabilities not detected

register_project(&mut self, project_input: ProjectInput)
Vulnerabilities not detected

update_project(&mut self, project_id: u64, project_input: ProjectInput)
Vulnerabilities not detected

publish_project(&mut self, project_id: u64)
Vulnerabilities not detected

unpublish_project(&mut self, project_id: u64)
Vulnerabilities not detected

get_project(&self, project_id: u64)
Vulnerabilities not detected

`get_projects(&self, from_index: Option<u64>, limit: Option<u64>)`
Vulnerabilities not detected

`get_projects_by_id(&self, project_ids: Vec<u64>)`
Vulnerabilities not detected

`get_num_balances(&self, project_id: u64, account_id: ValidAccountId)`
Vulnerabilities not detected

`remove_project(&mut self, project_id: u64)`
Vulnerabilities not detected

`project_deposit_in_token(&mut self, project_id: u64, amount: WrappedBalance)`
Vulnerabilities not detected

`project_withdraw_in_token(&mut self, project_id: u64, amount: Option<WrappedBalance>)`
Vulnerabilities not detected

`project_withdraw_in_token_and_astrodao(&mut self, project_id: u64)`
Vulnerabilities not detected

`project_withdraw_out_token(&mut self, project_id: u64, amount: Option<WrappedBalance>)`
Vulnerabilities not detected

`project_withdraw_remaining_out_token(&mut self, project_id: u64)`
Vulnerabilities not detected

Structure of contract

errors.rs

Contract methods analysis:

Vulnerabilities not detected

Structure of contract

utils.rs

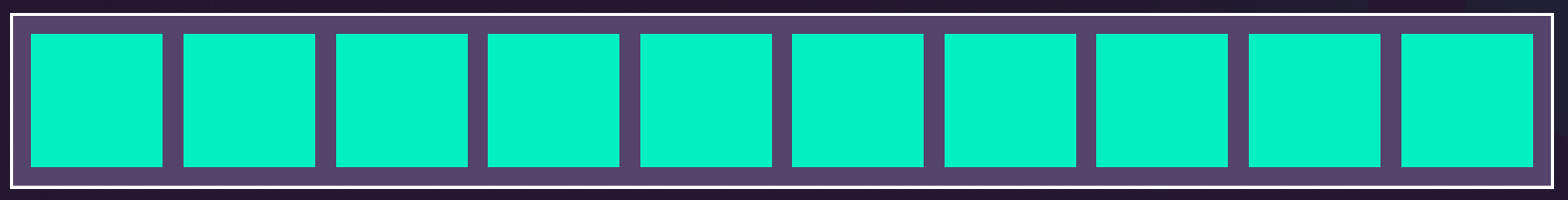
Contract methods analysis:

Vulnerabilities not detected

Verification check sums

Contract name	Bytecode hash(SHA 256)
internal.rs	4c0c08091da1ba0bbeceaf9a8380efe4ce0609150c4a4281dd26435e443545c6
lib.rs	4ead59cf12911f5107a07ae4608eddd31ae2dc38e1fb45c5ab68c0ad6420ae7f
project.rs	1cc4daf86365794a07dc0f89deeba0151ef420c65b6b773cb8b7aac482887673
errors.rs	1a36a3f571459fcbde0b1ba5eb622c14ac3a80364e3544c74e933ce1e955d6f2
utils.rs	99f9aa590bc3fcd4285f0a0511a2a1633b1e047f076d67dcf4170ab0a66383e6
Link to source code:	https://github.com/NearGenesis/launchpad-smartcontract

Project evaluation



10/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech