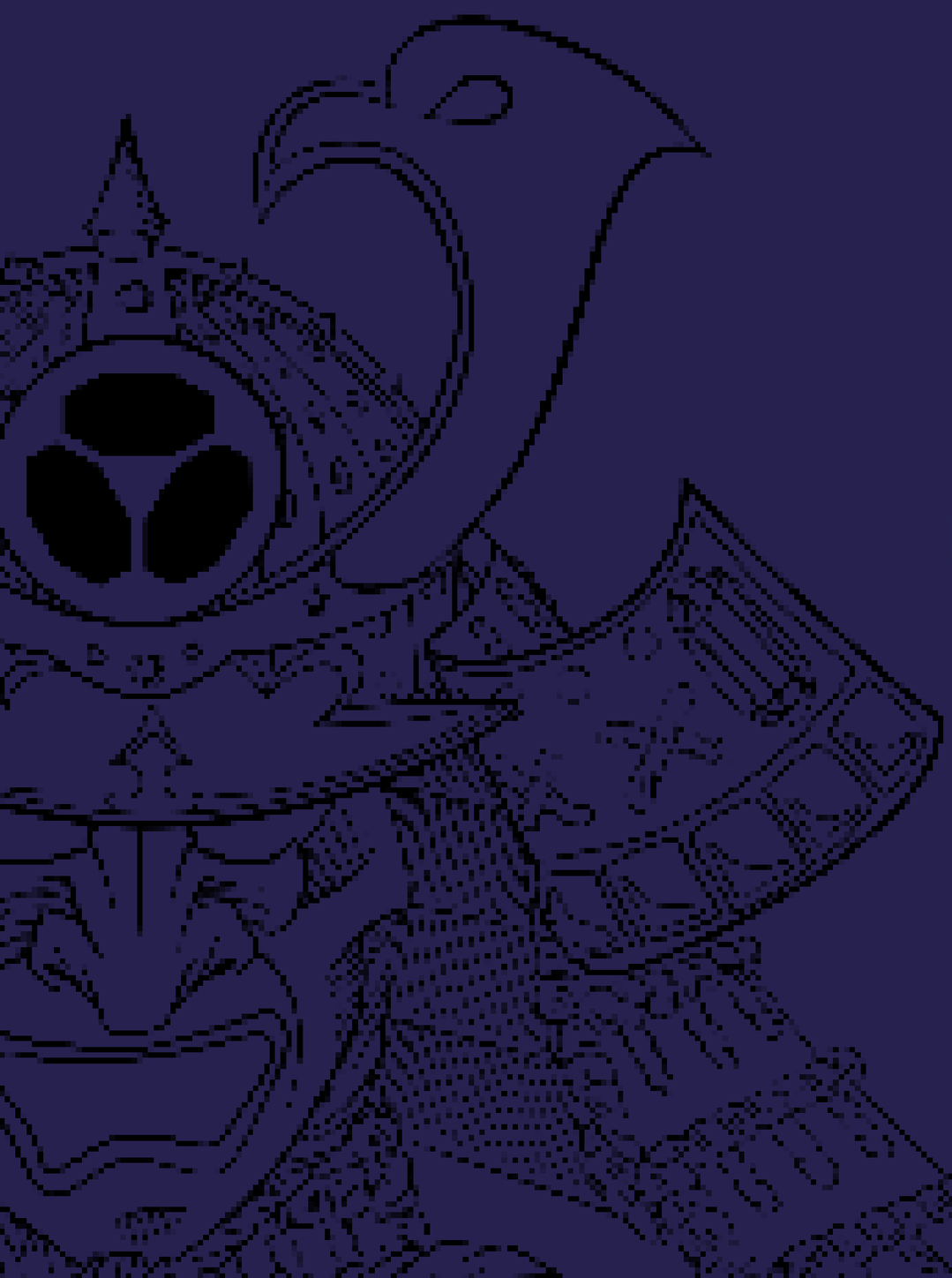

smart state

> Smart
Contract
Audit #

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unsed vars
- Uncheked 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

CONTRACT.RS

CONTRACT METHODS ANALYSIS:

- pub fn instantiate(deps: DepsMut, _env: Env, info: MessageInfo, msg: InstantiateMsg)
  Vulnerabilities not detected

- pub fn execute(deps: DepsMut, env: Env, info: MessageInfo, msg: ExecuteMsg)
  Vulnerabilities not detected

PAYABLE

- pub fn receive_cw20(deps: DepsMut,env: Env,info: MessageInfo,cw20_msg: Cw20ReceiveMsg)
  Vulnerabilities not detected

tokens in, public

- pub fn bond(deps: DepsMut, env: Env, sender_addr: Addr, amount: Uint128)
  User can bond even if staking period has ended, we would suggest to check for this case and not allow user to do this.

- pub fn unbond_init(deps: DepsMut, env: Env, info: MessageInfo, amount: Uint128)
  Vulnerabilities not detected

PAYABLE

- pub fn claim(deps: DepsMut, env: Env, info: MessageInfo)
  Vulnerabilities not detected

tokens out

- pub fn notify_rewards(deps: DepsMut,env: Env,info: MessageInfo,period_start: u64,period_finish: u64,amount: Uint128,unstaking_period: u64)
  Vulnerabilities not detected

PAYABLE

- pub fn recover_rewards(deps: DepsMut,env: Env,info: MessageInfo,amount: Uint128)
  Owner of the contract has an ability to take all the rewards from the contract via this method, so in case user would like to claim his rewards, he will claim from bonds of other users(because reward token and bond token are the same), because there are no free rewards to pay. Since the described scenario is protected by only owner policy the severity of this entry is on information level.

tokens out, only owner

- fn increase_bond_amount(state: &mut State, staker_info: &mut StakerInfo, amount: Uint128)
  Vulnerabilities not detected

- fn decrease_bond_amount(state: &mut State,staker_info: &mut StakerInfo,amount: Uint128,)
  Vulnerabilities not detected

- fn compute_state_extra(config: &Config, state: &mut State, timestamp: u64)
  Vulnerabilities not detected

- fn compute_reward(config: &Config, state: &mut State, timestamp: u64)
  Vulnerabilities not detected

- fn compute_staker_reward(state: &State, staker_info: &mut StakerInfo)
  Vulnerabilities not detected

- fn compute_claimable_amount(config: &Config,staker_info: &mut StakerInfo,timestamp: u64)
  Vulnerabilities not detected

- pub fn query(deps: Deps, _env: Env, msg: QueryMsg)
  Vulnerabilities not detected

- pub fn query_config(deps: Deps)
  Vulnerabilities not detected

- pub fn query_state(deps: Deps, timestamp: Option<u64>)
  Vulnerabilities not detected

- pub fn query_staker_info(deps: Deps,staker: String,timestamp: Option<u64>,)
  Vulnerabilities not detected

- pub fn migrate(_deps: DepsMut, _env: Env, _msg: MigrateMsg)
  Vulnerabilities not detected

STRUCTURE OF CONTRACT

QUERIER.RS

CONTRACT METHODS ANALYSIS:

- pub fn query_orion_balance(querier: &QuerierWrapper,orion_token: Addr,account: Addr)
  Vulnerabilities not detected

STRUCTURE OF CONTRACT

STATE.RS

CONTRACT METHODS ANALYSIS:

- pub fn store_config(storage: &mut dyn Storage, config: &Config)
  Vulnerabilities not detected
- pub fn read_config(storage: &dyn Storage)
  Vulnerabilities not detected
- pub fn store_state(storage: &mut dyn Storage, state: &State)
  Vulnerabilities not detected
- pub fn read_state(storage: &dyn Storage)
  Vulnerabilities not detected

- pub fn store_staker_info(storage: &mut dyn Storage,owner: &CanonicalAddr,staker_info: &StakerInfo,)
  Vulnerabilities not detected

- pub fn remove_staker_info(storage: &mut dyn Storage, owner: &CanonicalAddr)
  Vulnerabilities not detected

- pub fn read_staker_info(storage: &dyn Storage, owner: &CanonicalAddr)
  Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Bytecode hash (SHA 256)
Contract.rs	5c413271e889468c49177912003b4dd67043f3f2d79c87bf1f5e5fc9a3fb0573
State.rs	dc81a9dbe4f01508968f70d8fafa38654037929e49c5ab665d08afc92645591f
Querier.rs	ecde5e2724b37522ac25b214a28db9255c9910963d24b73dc820f7712a14cfe0

Get In Touch

info@smartstate.tech

smartstate.tech