AUDIT REPORT

TerraForm Labs Enterprise DAO

Prepared by SCV-Security

On 5th October 2023

Table of Contents

Introduction

SCV has been engaged by TerraForm Labs to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

Scope Functionality

Enterprise contracts aims to be a no-code tool for organising, creating, and maintaining Decentralised Autonomous Organizations (DAOs) on the Terra blockchain. With Enterprise, users can create multisig wallets, organise communities around NFTs and tokens, and manage the governance of DAOs within a single interface. Additionally, the ICS-Proxy contract allows Enterprise to move inter-chain by deploying the proxy-contract on designated operational chains. Enterprise makes the use of IBC and ICS specification.

Submitted Codebase

Enterprise DAO Contracts	
Repository	https://github.com/terra-money/enterprise-contracts
Branch	branch ics_attestation_audit
Commit	`3d00aaf79aec4ab9d77adb0828ed92931b5b1acd`
ICS-Proxy Contract	
Repository	https://github.com/terra-money/ics-proxy
Branch	ics_reply_mod
Commit	`9890cce82438fc61cf7a340668c2b101600df03a`

Revision Codebase

Enterprise DAO Contracts	
Repository	https://github.com/terra-money/enterprise-contracts
Branch	audit_fixes
Commit	`31eda2f599f08de045ff4c5ed2c3b42d32f0b518`
ICS-Proxy Contract	
Repository	https://github.com/terra-money/ics-proxy
Branch	ics_reply_mod
Commit	`a2167a302f0f7f5b9a5fb641595c61fb78cfbd5c`

Methodologies

SCV performs a combination of automated and manual security testing based on the scope of testing. The testing performed is based on the extensive experience and knowledge of the auditor to provide the greatest coverage and value to TerraForm Labs. Testing includes, but is not limited to, the following:

- Understanding the application and its functionality purpose.
- Deploying SCV in-house tooling to automate dependency analysis and static code review.
- Analyse each line of the code base and inspect application security perimeter.
- Review underlying infrastructure technologies and supply chain security posture.

Code Criteria

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organisation.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorised as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

Criteria	Status	Notes
Documentation	**SUFFICIENT**	N/A
Coverage	**NOT-SUFFICIENT**	Coverage tests were not sufficient. The audit submission had approximately 6% of coverage. It's recommended to considerably increase testing coverage around 60% to ensure core components are covered.
Readability	**NOT-SUFFICIENT**	The naming conventions for functions and associated variables lack clarity and descriptiveness, which impacts the readability of the codebase.
Complexity	**NOT-SUFFICIENT**	The Enterprise DAO utilizes an intricate multi-contract structure. It's essential to exert extra effort to minimize any unnecessary complexities within this design. In general, there's a need to simplify the codebase further.

Findings Summary

Summary Title	Risk Impact	Status
WeightsChanged message is permissionless, allowing attackers to manipulate voting weights	CRITICAL	RESOLVED
New members can claim more rewards with incorrect global index	CRITICAL	RESOLVED
Incorrect usage implemented for ComponentContracts query, breaking core functionalities	SEVERE	RESOLVED
Governance controller instantiation will fail due to unsaved DAO_TYPE storage	SEVERE	RESOLVED
Lack of delay when executing proposals makes contracts prone to governance attacks	SEVERE	PARTIALLY RESOLVED
Governance controller does not limit proposal creation	SEVERE	RESOLVED
create_proposal cannot support native funds	SEVERE	RESOLVED
Creation of poll will fail due to TOTAL_DEPOSITS not being initialized	SEVERE	RESOLVED
Existing council members are not migrated	SEVERE	RESOLVED
Instantiation of multi-sig membership contract fails	SEVERE	RESOLVED
Incorrect DaoType saved on membership creation	SEVERE	RESOLVED
Improper permissions will cause membership instantiation to fail	SEVERE	RESOLVED
Wrong assertion on sender for ibc_hook execution	SEVERE	RESOLVED
Updating CW20 and CW115 cross-chain treasury whitelists fails	SEVERE	RESOLVED
DAO type and unlocking period will not be stored in DAO_BEING_CREATED	SEVERE	RESOLVED
Treasury contract will not whitelist new instantiated CW20 or CW721 membership token address	SEVERE	RESOLVED
Enterprise contract instantiation fails because COMPONENT_CONTRACTS storage is read before stored	SEVERE	RESOLVED
distribute_funds will not distribute any funds	SEVERE	RESOLVED
end_proposal assumes that all proposals are General proposal types	SEVERE	RESOLVED
No entry point to send CW20 and CW721 tokens	SEVERE	RESOLVED

No auth validation when the whitelist is None	SEVERE	PARTIALLY RESOLVED
Instantiating proxy contract does not include the allow_cross_chain_msgs parameter	SEVERE	RESOLVED
Incorrect CW20 address prevents proposal execution	SEVERE	RESOLVED
Flaws in enterprise treasury contract migration	SEVERE	RESOLVED
Sending zero funds for old CW20 version might fail	MODERATE	RESOLVED
Treasury does not work with CW115 but supports it in add_whitelisted_assets_checked	MODERATE	RESOLVED
Proposal actions are unbounded	MODERATE	RESOLVED
Inefficient migration of asset tokens	MODERATE	RESOLVED
Proposals cannot be created if a minimum deposit is applied to Denom DAO type	MODERATE	RESOLVED
Fund distributor is not set with initial members correctly	MODERATE	ACKNOWLEDGED
query_council_total_weight uses the wrong response struct	MODERATE	RESOLVED
DAO can be locked if there are no initial members	MODERATE	RESOLVED
Missing treasury SetAdminMsg message	MODERATE	ACKNOWLEDGED
Attacker may interfere with funds distributor by sending many assets	MODERATE	RESOLVED
Use serde_cw_value::Value instead of serde_json::Value	LOW	RESOLVED
QueryMsg::Config calls the wrong query function	LOW	RESOLVED
Governance controller has no way of handling failed proposal executions	LOW	ACKNOWLEDGED
Validate_proposal_actions does not properly validate RequestFundingFromDao	LOW	RESOLVED
Duplicated unlocking_period on create_dao	LOW	RESOLVED
Unreachable CW20 Hook variants	LOW	ACKNOWLEDGED
claim_rewards allows the caller to initiate a claim for any user	LOW	RESOLVED
DistributeNative submessage will be added even if native_funds is empty	LOW	RESOLVED
Missing duplicate validation in initialize_stakers	LOW	RESOLVED
Reduce gas usage using max limit instead of default limit in get_versions_between_current_and_target	INFO	RESOLVED

Gas usage can be reduced in import_cw3_membership function	INFO	RESOLVED
Event emitting wrong response	INFO	RESOLVED
Gas usage can be reduced by breaking out of the loop early	INFO	RESOLVED
add_cross_chain_proxy function emits incorrect response	INFO	RESOLVED
Incorrect label when instanting enterprise contract	INFO	RESOLVED
Rename variable for clarity	INFO	RESOLVED
user_stake is not emitted	INFO	RESOLVED
Council membership and attestation address are not emitted	INFO	RESOLVED
Query performed directly when associated function exists	INFO	RESOLVED
Proposal UpdateCouncil can update only all fields	INFO	ACKNOWLEDGED
ENTERPRISE_CODE_IDS is unused	INFO	ACKNOWLEDGED
Proxy contract does not implement queries	INFO	RESOLVED

Findings Technical Details

1. `WeightsChanged` message is permissionless, allowing attackers to manipulate voting weights

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `weights_changed` function in `contracts/enterprise-governance-controller/src/contract.rs:1151` is permissionless and allows any caller to update user weights and user votes. This is problematic because an attacker can call this function to artificially change the voting weights, allowing malicious proposals to be executed.

Recommendation

We recommend updating the `weights_changed` function to validate the `info.sender` to be the membership contract.

2. New members can claim more rewards with incorrect global index

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

In the `claim_rewards` function, users that do not have `EFFECTIVE_USER_WEIGHTS` can update their global index in `NATIVE_DISTRIBUTIONS` and `CW20_DISTRIBUTIONS` even though they do not have a valid stake.

If the user has `NATIVE_DISTRIBUTIONS` and `CW20_DISTRIBUTIONS` storage set, `update_user_weights` will not update to the latest global index in `contracts/funds-distributor/src/user_weights.rs:152` and line 174.

This allows users to claim more rewards for periods they did not stake, causing other users to not receive any funds.

Please refer to this Gist link to reproduce the vulnerability..

Recommendation

We recommend disabling the user from claiming rewards when `EFFECTIVE_USER_WEIGHTS` for them is zero.

3. Incorrect usage implemented for `ComponentContracts` query, breaking core functionalities

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `QueryMsg::ComponentContracts` query in `contracts/enterprise/src/contract.rs:388` incorrectly calls the `query_dao_info` function instead of the intended `query_component_contracts` function. This causes other contract integrations to fail because an incorrect response is returned.

The following functions are affected because they rely on the `ComponentContracts` query to get the contract addresses:

- `contracts/enterprise-governance-controller/src/contract.rs:728`
- `packages/membership-common/src/validate.rs:38`
- `contracts/enterprise-governance-controller/src/contract.rs:1169`
- `contracts/enterprise-governance-controller/src/contract.rs:1725`

Recommendation

We recommend updating the `ComponentContracts` query to call the `query_component_contracts` function in `contracts/enterprise/src/contract.rs:414` and return `ComponentContractsResponse`.

4. Governance controller instantiation will fail due to unsaved DAO_TYPE storage

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When instantiating the `enterprise-governance-controller`, a `DaoInfo` query is dispatched in `contracts/enterprise-governance-controller/src/contract.rs:133` to the previously initialized enterprise contract. However, `DAO_TYPE` is only saved after executing the `FinalizeInstantiation` message. Attempting to query `DaoInfo` before finalization will result in failure since it has not been finalized, ultimately causing the instantiation of the `enterprise-governance-controller` contract to fail.

Recommendation

We recommend saving the `DAO_TYPE` during the instantiation of the enterprise contract rather than later during the `FinalizeInstantiation` execution, or using the `DAO_TYPE` that can be found in the `DAO_BEING_CREATED` struct.

5. Lack of delay when executing proposals makes contracts prone to governance attacks

RISK IMPACT: SEVERE	STATUS: PARTIALLY RESOLVED

Revision Notes

The team has mentioned that they have effectively resolved the matter concerning early-executable proposals and execution delays. Nonetheless, they have yet to tackle the issue that prevents proposals from being executed after their specified ending time. This specific matter demands additional scrutiny, particularly in terms of managing deposits and its potential implications for future use cases involving Warp.

Description

The `end_proposal` function in `contracts/enterprise-governance-controller/src/contract.rs:496` does not enforce any time delays between a proposal passing and being executed. This could allow a governance attack for proposals that are allowed to end early:

1. An attacker stakes a large number of tokens to gain significant voting power.
2. The attacker creates a malicious proposal and immediately votes for it with their voting power.
3. If the proposal passes the vote threshold, it will be immediately executed without any delay since `end_proposal` does not enforce time delays.
4. This would allow the attacker to exploit the lack of a time delay window to execute their attack before the community has time to react and block it.

Recommendation

We recommend enforcing a delay period in `end_proposal` between a proposal passing and being executed. Additionally, the proposal should be disallowed to be executed if it passes a specific period.

6. Governance controller does not limit proposal creation

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `CreateProposal` `ExecuteMsg` in `contracts/enterprise-governance-controller/src/contract.rs:158` allows a caller to create a proposal with no deposit as long as the `gov_config.minimum_deposit` is `None`. If it happens, this will remove the mechanism for spam prevention allowing for spam proposals to be created for any DAO type.

During the instantiation of the governance controller, the `validate_dao_gov_config` function should ensure that the `minimum_deposit` is `Some`. Even if a Dao Type is `NFT`, a base minimum deposit should be enforced to limit spam and low-quality proposals. While the current design of the protocol does seem to support the ability to create proposals without deposits, it is best practice to implement a non-zero minimum deposit to prevent spam.

Recommendation

We recommend defining a non-zero `minimum_deposit` regardless of the dao type and ensuring that this amount is enforced in the `validate_dao_gov_config` function. We do note that if the `Multisig` membership contract is deemed to be sufficiently restrictive by its membership requirements and small size, that type does not explicitly require a proposal deposit. Additionally, it is important to note that the `create_proposal` function does not properly handle funds *create_proposal cannot support native funds* to fully resolve this issue both remediations must be handled.

7. `create_proposal` cannot support native funds

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `create_proposal` function in `contracts/enterprise-governance-controller/src/contract.rs:170` is not properly configured to handle native funds sent in the `ProposalDeposit` for `Denom` DAO type. While this feature is currently not implemented and is hard-coded as `None`, proposal deposits serve as a critical component in controlling spam as discussed in *Governance controller does not limit proposal creation*. To resolve the issues described in the finding will require that the hard-coded amount be replaced.

Recommendation

To enable the deposit functionality we recommend implementing the following features in conjunction with the finding *Governance controller does not limit proposal creation*:

- Validate that `info.funds` is equal to the non-zero minimum deposit
- Validate the dao type is not `Token` as this is handled by the `receive_cw20` function.

8. Creation of poll will fail due to `TOTAL_DEPOSITS` not being initialized

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

In the `CREATE_POLL_REPLY_ID` in `contracts/enterprise-governance-controller/src/contract.rs:1333`, the `TOTAL_DEPOSIT` is never initialized but directly updated. Consequently, when a proposal is created and tokens have been deposited, it triggers a `NotFound` error due to the absence of an initial value for `TOTAL_DEPOSIT`, causing the creation of the poll to fail.

Recommendation

We recommend saving the value of `TOTAL_DEPOSIT` during contract instantiation.

9. Existing council members are not migrated

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `create_governance_controller_contract` function in `contracts/enterprise-treasury/src/migration.rs:345` sets the council members into an empty vector instead of the intended council members. This is inconsistent with the DAO council membership contract as it instantiates with all valid members in line `290`.

Consequently, there will be a mismatch between the council governance members in the governance controller contract and the actual members in the DAO council membership contract.

Recommendation

We recommend storing the members as `council.members` instead of an empty vector.

10. Instantiation of multi-sig membership contract fails

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `save_initial_weights` function in `packages/multisig-membership-impl/src/instantiate.rs:29` is invoked during contract instantiation. However, there is an issue with the `load_total_weight` function when the `msg.initial_weights` parameter is provided as `Some(_)`. Specifically, it attempts to load the `TOTAL_WEIGHT_HEIGHT_SNAPSHOT` storage which is never stored before. Consequently, this loading operation will fail, leading to a failure in the contract's initialization process.

Recommendation

We recommend using `unwrap_or_default` on the `load_total_weight`.

11. Incorrect DaoType saved on membership creation

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When instantiating the membership contract in `contracts/enterprise-factory/src/contract.rs:490`, the `instantiate_denom_staking_membership_contract` function incorrectly sets the DAO type to `Token` in `contracts/enterprise-factory/src/denom_membership.rs:22`. This is incorrect because the value should be `Some(DaoType::Denom)`.

Recommendation

We recommend updating the DAO type to `DaoType::Denom`.

12. Improper permissions will cause membership instantiation to fail

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `AddWeightChangeHook` message is constructed by the Enterprise Factory contract to be passed to the Membership contract during instantiation within the `MEMBERSHIP_CONTRACT_INSTANTIATE_REPLY_ID` and `COUNCIL_MEMBERSHIP_CONTRACT_INSTANTIATE_REPLY_ID` reply handlers in `contracts/enterprise-factory/src/contract.rs:335` and 365.

This is problematic because the `add_weight_change_hook` function in the membership contracts requires that the caller be the governance controller contract and this is enforced by the `enterprise_governance_controller_only` function in `packages/membership-common/src/weight_change_hooks.rs:18`. However, since the message sender is the Factory contract rather than the Governance Controller contract, this authorization check will fail which will cause an error during the instantiation phase.

Recommendation

We recommend updating the membership contract instantiation process to include a message to allow the initial weight change hooks to be set by the factory contract strictly during the instantiation.

13. Wrong assertion on sender for `ibc_hook` execution

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When instantiating a new proxy contract via IBC hooks, the owner and whitelist addresses are set to `env.contract.address` in `contracts/enterprise-governance-controller/src/contract.rs:1060` and `1061`. This value is asserted on each proxy contract call to ensure the caller is authorized in `icd-proxy-icd_reply_mod/src/contract.rs:94, 200,` and `243`. However, the IBC hook packet sender field is an untrusted field as described in the following documentation:

"We *cannot trust the sender of an IBC packet, the counterparty chain has full ability to lie about it. We cannot risk this sender being confused for a particular user or module address on Osmosis. So we replace the sender with an account to represent the sender prefixed by the channel and a wasm module prefix. This is done by setting the sender to Bech32(Hash("ibc-wasm-hook-intermediary" || channelID || sender)), where the channelId is the channel id on the local chain. This will make any ibc-hook transaction fail.*"

This will cause the proxy owner assertion to fail, preventing the IBC hook from executing the intended cross-chain logic.

Recommendation

Precompute the `ibc-hook` address that will be assigned to the `info.sender` during contract execution and save this address instead of `env.contract.address` on line `1060` & `1061`. Apply the same during the execution of `ExecuteMsgReplyCallback` on controller, assert the `info.sender` creating the `bench32` address using `ibc-hook` logic and the proxy address.
It should be taken into consideration that the PFM cannot be used, as the address that is assigned in the info.sender during the execution of the message via `ibc-hook` will be further different

14. Updating CW20 and CW115 cross-chain treasury whitelists fails

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The `validate_asset_whitelist_changes` function in `contracts/enterprise-governance-controller/src/validate.rs:214` and line `215` calls the `split_asset_hashsets` function to validate the CW20 and CW1155 token addresses.

This means that the CW20 and CW1155 tokens will be local chain token contracts due to the `addr_validate` function. However, this validation would prevent cross-chain token contracts from being whitelisted successfully because the local token contracts do not work on remote chains. The message will be dispatched to the remote chain in `contracts/enterprise-governance-controller/src/contract.rs:812` when `msg.remote_treasury_target` is `Some`.

Recommendation

We recommend not performing the validation if `msg.remote_treasury_target` is `Some` because the expected token contracts are for remote chains.

15. DAO type and unlocking period will not be stored in DAO_BEING_CREATED

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

In `contracts/enterprise-factory/src/contract.rs:525`, `DAO_BEING_CREATED` saves the governance controller address and fills the remaining fields with the old `dao_being_created` variable created in line 417.

This would overwrite the changes made by the `instantiate_new_cw20_membership` and `instantiate_new_cw721_membership` functions in `contracts/enterprise-factory/src/token_membership.rs:56-57` and `contracts/enterprise-factory/src/nft_membership.rs:42-43`.

Consequently, the DAO type and the unlocking period will not be stored, causing the instantiation process to fail in line `contracts/enterprise-factory/src/contract.rs:279` and line 304.

Recommendation

We recommend using the `update` function instead of the `save` function at `contracts/enterprise-factory/src/contract.rs:519`.

16. Treasury contract will not whitelist new instantiated CW20 or CW721 membership token address

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

During the instantiation process, `dao_being_created.dao_nft` and `dao_being_created.dao_asset` are both saved in lines `contracts/enterprise-factory/src/contract.rs:284` and `309` after instantiating the CW20 or CW721 membership contract. The `dao_nft` or `dao_asset` will be used in lines `465` and `470` when instantiating the enterprise treasury contract.

However, since instantiating the CW20 or CW721 addresses happens after the mentioned lines above, the addresses won't be stored properly. Specifically, the CW20 and CW721 will be instantiated in lines `496` and `498` respectively, but the values are already used in lines `465` and `470`.

Consequently, the asset and NFT whitelist for the treasury contract will not include the newly instantiated CW20 or CW721 token addresses.

Recommendation

We recommend instantiating the CW20 or CW721 token contracts before instantiating the treasury contract so the treasury contract includes them as whitelists.

17. Enterprise contract instantiation fails because COMPONENT_CONTRACTS storage is read before stored

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

In the `add_weight_change_hook` function in `packages/membership-common/src/weight_change_hooks.rs:18`, the `enterprise_governance_controller_only` is called to ensure the caller is the governance controller contract. Underlying the hook, it dispatches a `ComponentContracts` query message to the enterprise contract to get the governance controller address, as seen in `packages/membership-common/src/validate.rs:35-38`.

The issue lies where the query is executed before the enterprise contract stores the addresses in the `COMPONENT_CONTRACTS` storage. The enterprise factory contract instantiates all contracts first before finalizing the enterprise contract. Once the enterprise contract is finalized, the `COMPONENT_CONTRACTS` storage will be stored in `contracts/enterprise/src/contract.rs:138`. The storage is used when the `query_component_contracts` function is called in `contracts/enterprise/src/contract.rs:415`.

During the instantiation process, the `add_weight_change_hook` function will be called in `contracts/enterprise-factory/src/contract.rs:335` to register the hook. However, when the `enterprise_governance_controller_only` function is called, the `ComponentContracts` query will fail because the `COMPONENT_CONTRACTS` storage in the enterprise contract is not set yet because the enterprise contract is not finalized yet. Consequently, the enterprise contract instantiation process will fail.

Recommendation

We recommend modifying the `enterprise_governance_controller_only` function to not rely on the unsaved `COMPONENT_CONTRACTS` storage to get the enterprise governance controller address before the enterprise contract is fully instantiated.

18. `distribute_funds` will not distribute any funds

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

In the `distribute_funds` function in `contracts/enterprise-treasury/src/contract.rs:196`, a `Response` object is created and returned, but the sub-messages generated earlier are not included in the response. This results in the sub-messages not being dispatched, which breaks the intended functionality of distributing funds.

Recommendation

We recommend modifying the `distribute_funds` function to include the sub-messages in the response.

19. `end_proposal` assumes that all proposals are General proposal types

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When ending a proposal in the `end_proposal` function in `contracts/enterprise-governance-controller/src/contract.rs:506`, the `total_available_votes` function incorrectly assumes that all votes are `General` `proposal_type`. This is incorrect as proposals created through the `CreateCouncilProposal` message are `Council` proposal type, as seen in `contracts/enterprise-governance-controller/src/contract.rs:258`. Consequently, the computed total votes will be incorrect, causing proposals to be evaluated incorrectly.

Recommendation

We recommend using the `query_council_total_weight` function to evaluate the total available votes for the `Council` proposal type.

20. No entry point to send CW20 and CW721 tokens

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When receiving CW20 and CW721 tokens in `packages/token-staking-impl/src/execute.rs:39` and `packages/nft-staking-impl/src/execute.rs:34`, the `receive_cw20` and `receive_nft` functions validates the `msg.sender` to ensure to be the governance controller contract.

However, there are no available entry points within the governance controller contract that facilitate the sending of CW20 or CW721 tokens. This results in an inability for any entity to transfer CW20 and CW721 tokens to these contracts.

Consequently, the usability of the contract will be affected as users cannot call the receive entry points to stake their tokens or NFT.

Recommendation

We recommend updating these entry points to remove the validation that restricts the caller to be only the governance controller.

21. No auth validation when the whitelist is None

RISK IMPACT: SEVERE	STATUS: PARTIALLY RESOLVED

Revision Notes

The team has underlined the importance of enabling proxy contracts to execute `execute_msgs` without permission restrictions. This is a crucial requirement because the team has plans to deploy a universal permissionless proxy on each chain which will serve as a foundation for DAOs to create their own permissioned proxies. A recent improvement in this regard ensures that the owner is added to the whitelist, even when the whitelist is `None`.

Description

The `execute_msgs` function in `src/contract.rs:91` of the ICS proxy repository does not validate the caller if there is no whitelist. Anyone can execute any message as long it adheres to the `allow_cross_chain_msgs` parameter.

Additionally, in `contracts/enterprise-governance-controller/src/contract.rs:1111`, the proxy contract is the admin of the treasury contract. If the proxy contract does not have any whitelist, anyone can control the proxy contract to call the treasury contract to steal funds using the `Spend` or `DistributeFunds` messages.

Recommendation

We recommend returning an error if the whitelist is `None`.

22. Instantiating proxy contract does not include the `allow_cross_chain_msgs` parameter

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

When a proxy contract is instantiated, the `allow_cross_chain_msgs` parameter is required in `src/contract.rs:49` of the ICS proxy repository. However, the enterprise governance controller does not include the parameter when instantiating a cross-chain proxy. In `contracts/enterprise-governance-controller/src/contract.rs:1059-1063`, the `IcsProxyInstantiateMsg` struct does not include the `allow_cross_chain_msgs` parameter, causing the instantiation of the proxy contract to fail.

Recommendation

We recommend including the `allow_cross_chain_msgs` parameter in the `IcsProxyInstantiateMsg` struct.

23. Incorrect CW20 address prevents proposal execution

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

In `contracts/enterprise-governance-controller/src/contract.rs:485`, the `membership_contract` variable is being used as the CW20 token address when validating token membership proposals. However, this address corresponds to the `token-staking-membership` contract, not the actual CW20 token contract. This will prevent token membership proposals from being properly ended. The token's CW20 address needs to be used instead of the `membership_contract` address.

Recommendation

We recommend using the CW20 token address by retrieving the `token_contract` value using the `query_token_config` query.

24. Flaws in enterprise treasury contract migration

RISK IMPACT: SEVERE	STATUS: RESOLVED

Description

The current configuration of the treasury contracts migration is flawed and will not execute as intended. This is because the function first transfers admin privileges to the enterprise contract, and then later during the execution will attempt to migrate the contract of which it is no longer an admin.

In `contracts/enterprise-treasury/src/migration.rs:158`, two sub-messages are prepared for dispatch. However, an inconsistency arises in the sequence of operations. The logic associated with the first reply, identified as `ENTERPRISE_INSTANTIATE_REPLY_ID`, updates the contract's administrative rights to the enterprise contract. Following this, a subsequent message, `FinalizeMigration`, invokes the migrate function on these contracts. This action will fail, given that the admin of the contracts has been previously updated to enterprise in the earlier reply handler.

In `contracts/enterprise-treasury/src/migration.rs:582`, the `FinalizeInstantiation` message is intended to be executed on the enterprise contract, but this will not work as intended because the sender is checked to only be the factory contract in `contracts/enterprise/src/contract.rs:108`. The value is set to `ENTERPRISE_FACTORY_CONTRACT` when the enterprise contract is instantiated in `contracts/enterprise-treasury/src/migration.rs:145` during the migration. This would fail and block the migration.

Recommendation

We recommend updating the enterprise treasury migration to perform the migration first and then transfer the admin permission to the enterprise contract.

25. Sending zero funds for old CW20 version might fail

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

In the `governance_controller_contract_created` function in `contracts/enterprise-treasury/src/migration.rs:398`, if the `deposit_amount` is zero in the `send_deposits_submsg`, the `transfer_msg` will fail for legacy CW20 versions because sending 0 amounts are not allowed.

Recommendation

We recommend only performing `transfer_msg` if the amount is not 0.

26. Treasury does not work with CW115 but supports it in `add_whitelisted_assets_checked`

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `enterprise-treasury` contract currently does not support CW115 asset type, yet in the `add_whitelisted_assets_checked` function in `contracts/enterprise-treasury/src/asset_whitelist.rs:34` it supports `AssetInfo::CW115`.

This means that even if the CW1155 asset was added, it cannot be distributed, causing it to be locked in the contract.

Recommendation

We recommend removing the support for the CW115 asset type.

27. Proposal actions are unbounded

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `validate_proposal_actions` function in `contracts/enterprise-governance-controller/src/validate.rs:108` does not enforce any limit on the number of total proposal actions that can be specified in one proposal. This can create a situation where a proposal can pass, but then its messages will never be able to be executed with `execute_proposal_actions_submsgs` due to excessive gas costs for the proposal executor.

Recommendation

Consider implementing a validation to enforce a maximum number of proposal actions that a single proposal can contain.

28. Inefficient migration of asset tokens

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `map_whitelisted_assets` function at `contracts/enterprise-treasury/src/migration.rs:161` performs an inefficient migration process. It loads all assets from `NATIVE_ASSET_WHITELIST`, `CW20_ASSET_WHITELIST`, and `CW1155_ASSET_WHITELIST`, only to save them back to the same storage using the `add_whitelisted_assets_checked` function. These same migrated assets are subsequently removed in `contracts/enterprise-treasury/src/migration.rs:596-598`.

As a result, the previous state migrations appear redundant since they don't persist after migration. This is because the latest version still utilizes the same storage, as observed in `contracts/enterprise-treasury/src/asset_whitelist.rs:8-10`.

Recommendation

We recommend removing the `map_whitelisted_assets` function during migration and avoiding clearing the old storage in `contracts/enterprise-treasury/src/migration.rs:596-598` such that old assets configured are still retained after migration.

29. Proposals cannot be created if a minimum deposit is applied to Denom DAO type

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `create_proposal` function in `contracts/enterprise-governance-controller/src/contract.rs:190` validates the deposit amount to the `gov_config.minimum_deposit` value. The entry points come from `ExecuteMsg::CreateProposal` with the deposit set to `None` and from `Cw20HookMsg::CreateProposal` which supports CW20 proposal deposits.

The issue happens when the DAO type is `Denom` and a minimum deposit is applied. Since there is no entry point for the user to satisfy the minimum deposit amount, proposals cannot be created properly.

Recommendation

We recommend adding support for the `Denom` DAO type to satisfy the minimum deposit requirement.

30. Fund distributor is not set with initial members correctly

RISK IMPACT: MODERATE	STATUS: ACKNOWLEDGED

Revision Notes

The team has clarified that this isn't a concern because only multisig memberships come with initial weights, while the other types rely on staking. Since there are no stakes involved when creating new contracts, the initial weights for multisig memberships are correctly configured.

Description

In the `ENTERPRISE_GOVERNANCE_CONTROLLER_INSTANTIATE_REPLY_ID` reply handler, the initial weights in `contracts/enterprise-factory/src/contract.rs:424` will be an empty vector because it is `None`. This means the fund distributor contract will be instantiated with empty members in `line 442`.

The fund distributor contract should be registered with all members from the DAO membership contract and council member contract. However, the initial members are not updated to the fund distributor contract after registering the hook address with the `AddWeightChangeHook` message.

Consequently, only the updated members will have their weight updated in the fund distributor contract. This is incorrect as members who don't have their weight updated are not reflected there. The same applies to the council members.

For example, if the DAO membership uses `ImportCw3` in line `499`, the members from the existing multi-sig contract do not have their power reflected in the fund distributor contract.

Recommendation

We recommend calling the `WeightsChanged` message after instantiating to register the initial members correctly for the DAO membership and council membership contract.

31. query_council_total_weight uses the wrong response struct

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

In the `query_council_total_weight` function in `contracts/enterprise-governance-controller/src/contract.rs:1782`, the query for `member_weight` is set to `UserWeightResponse` when it should be `TotalWeightResponse` instead, causing the query to fail.

Recommendation

We recommend replacing `UserWeightResponse` with `TotalWeightResponse`.

32. DAO can be locked if there are no initial members

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

In the `instantiate_new_cw20_membership` function in `contracts/enterprise-factory/src/token_membership.rs:100`, setting empty vector for `msg.initial_token_balances` and `msg.token_mint` to `None` causes DAO to be locked.

Similarly, in the `instantiate_new_multisig_membership` function `contracts/enterprise-factory/src/contract.rs:500`, If the user weight provided in `NewMultisig` is empty, DAO will be locked.

The reason behind this is that there are no initial token owners or members for the DAO. This means that no one will have voting power to control the DAO to mint new tokens or issue proposals, locking the DAO functionality.

Recommendation

We recommend validating that the above values are not instantiated with a 0 initial weight and empty vector members.

33. Missing treasury `SetAdminMsg` message

RISK IMPACT: MODERATE	STATUS: ACKNOWLEDGED

Revision Notes

The team has mentioned that they may consider employing the `SetAdmin` function for cross-chain deployments in the future. This would allow them to deploy new proxies while retaining the existing treasury if the need arises.

Description

The treasury contract implements a `SetAdminMsg` that is used to update the admin value in `contracts/enterprise-treasury/src/contract.rs:87`. Typically, this message should be sent by the current admin, which, in this case, is the governance controller. However, the governance controller does not define a `SetAdminMsg`, making it impossible to access this entry point.

Recommendation

We recommend either implementing the `SetAdminMsg` on the governance controller, or remove the `SetAdminMsg` on the treasury if it is not intended to be executed.

34. Attacker may interfere with funds distributor by sending many assets

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

In the `query_voter` function in `packages/poll-engine/src/query.rs:140`, the query involves an unbounded iteration over all the votes made by a specific voter. In scenarios where there are many proposals, and the voter voted for most of them, this query may fail due to an out-of-gas error.

In addition, the `initialize_user_indices`, `update_user_native_distributions`, and `update_user_cw20_distributions` functions perform an unbounded iteration through the `NATIVE_GLOBAL_INDICES` and `CW20_DISTRIBUTIONS` storage in the following lines:

- `contracts/funds-distributor/src/user_weights.rs:137`
- `contracts/funds-distributor/src/user_weights.rs:159`
- `contracts/funds-distributor/src/native_distributions.rs:61`
- `contracts/funds-distributor/src/cw20_distributions.rs:61`

An attacker can cause the `UpdateUserWeights` message to fail by spamming the `NATIVE_GLOBAL_INDICES` and `CW20_DISTRIBUTIONS` storage with many fake tokens. Specifically, an attacker can create factory token denoms using the `x/tokenfactory` module and add the denoms using the `DistributeNative` message. For CW20 tokens, an attacker can instantiate several CW20 token contracts and add them using the `Cw20HookMsg::Distribute` message.

If the iterations for `NATIVE_GLOBAL_INDICES` and `CW20_DISTRIBUTIONS` storage become too large, an out-of-gas error might occur, causing the transaction to fail.

Recommendation

We recommend implementing a whitelist for allowed native tokens and CW20 tokens.

35. Use serde_cw_value::Value instead of serde_json::Value

RISK IMPACT: LOW	STATUS: RESOLVED

Description

The function `serde_json::from_slice` used in the dao upgrade functionality in `contracts/enterprise-governance-controller/src/validate.rs:344` and `contracts/enterprise/src/contract.rs:204` accepts a `[u8]` value and attempts to deserializes it into a `serde_json::Value`. The problem arises during deserialization because `serde_json::Value` checks for the presence of an f64 number. When the Cosmwasm virtual machine detects this f64 operator, it halts the process of loading the code onto the blockchain. This will cause the migrate_msg_json to fail which will ultimately throw an error in `validate_upgrade_dao`.

Recommendation

We recommend using `serde_cw_value::Value` instead of `serde_json::Value`.

36. `QueryMsg::Config` calls the wrong query function

RISK IMPACT: LOW	STATUS: RESOLVED

Description

In the `parse_poll_id` function in `contracts/enterprise-governance-controller/src/contract.rs:1413`, `QueryMsg::Config` calls the `query_gov_config` function instead of the `query_config` function.

Recommendation

We recommend modifying the query to call the corresponding function.

37. Governance controller has no way of handling failed proposal executions

RISK IMPACT: LOW	STATUS: ACKNOWLEDGED

Revision Notes

The team has mentioned that they desire to enhance error handling, particularly in terms of providing more informative error messages. Currently, when it comes to contract-to-contract errors in CosmWasm, the errors are non-verbose and only consist of error codes, making it challenging to identify the specific error encountered.

Description

The `enterprise-governance-controller` contract does not have any way of gracefully handling proposals that pass but fail to execute. In `contracts/enterprise-governance-controller/src/contract.rs:573` proposal actions messages are dispatched with `reply_always`. For `ReplyOn::Always` submessages and `ReplyOn::Error` the individual submessage will error but it will not revert the entire transaction. It is the responsibility of the calling contract to handle the reply within its reply entrypoint.

Currently the `EXECUTE_PROPOSAL_ACTIONS_REPLY_ID` reply id is a no-op. This means that a proposal fails to execute without its failure being handled.

Recommendation

We recommend implementing logic in the reply handler to represent an error state that the proposal may be in if its execution has failed.

38. `Validate_proposal_actions` does not properly validate `RequestFundingFromDao`

RISK IMPACT: LOW	STATUS: RESOLVED

Description

The `validate_proposal_actions` function in `contracts/enterprise-governance-controller/src/validate.rs:136` does not properly validate the `RequestFundingFromDao` proposal action. This can present a situation where a proposal with a `RequestFundingFromDao` action passes but then its execution will fail due to the unvalidated message.

There are two major validations that are missing from the function and may be problematic. The assets vector is unchecked, meaning that any invalid asset will prevent the entire proposal from being executed. Additionally, in the current scope of the audit, the treasury does not support spending the `CW1155` asset type, this means that even if all the assets supplied are valid, the proposal will still fail to execute as that type is not supported for the treasury to spend. Additionally, the recipient field of the type is also not validated.

For any functionality related to proposals, it is best practice to pre-validate any possible conditions that may cause the proposal message execution to fail. If a proposal passes voting initially but then it fails to execute, there is no guarantee that the voting outcome will be the same if it is proposed again.

Recommendation

We recommend adding the validations mentioned above to the `validate_proposal_actions` function.

39. Duplicated `unlocking_period` on `create_dao`

RISK IMPACT: LOW	STATUS: RESOLVED

Description

The `CreateDaoMsg` contains two instances of `unlocking_period`. One is defined in `GovConfig` in `packages/enterprise-governance-controller-api/src/api.rs:40`, and the other is defined on any `CreateDaoMembershipMsg` variant. Notably, `GovConfig.unlocking_period` is never actually used in a `dao_creation`.

Recommendation

We recommend removing the `unlocking_period` from `GovConfig`.

40. Unreachable CW20 Hook variants

RISK IMPACT: LOW	STATUS: ACKNOWLEDGED

Revision Notes

The team has mentioned that is a more substantial issue that will require a significant amount of additional code. The implementation of these hooks serves the specific purpose of effectively addressing the migration challenge related to the incomplete transfer of stakes and claims to token and NFT contracts.

Description

The `receive_cw20` function in `packages/token-staking-impl/src/execute.rs:30` defines two `Cw20HookMsg` variants that are not used and are unreachable as only the governance controller can call the function but the governance controller does not define the message variants.

Other unused hook variants are:

- `packages/nft-staking-impl/src/execute.rs:38`,
- `packages/token-staking-impl/src/execute.rs:43`,
- `packages/token-staking-impl/src/execute.rs:44`.

Recommendation

We recommend removing the `InitializeStakers`, `AddClaims`, and `Cw20HookMsg` variants.

41. `claim_rewards` allows the caller to initiate a claim for any user

RISK IMPACT: LOW	STATUS: RESOLVED

Description

The `claim_rewards` in `contracts/funds-distributor/src/claim.rs:21` function allows the caller to claim rewards for any user. The `ClaimRewardsMsg` allows the caller to specify any address as `msg.user`, allowing a caller to initiate a rewards claim for a user that does not intend to claim rewards at the time for any reason such as tax implications, etc.

This is also present in the `claim` function in `packages/denom-staking-impl/src/execute.rs:130`.

Recommendation

We recommend verifying that the `info.sender` is equal to the user address that the claim is being made for.

42. DistributeNative submessage will be added even if `native_funds` is empty

RISK IMPACT: LOW	STATUS: RESOLVED

Description

The `distribute_funds` function builds up a submsgs vector containing calls to distribute assets. For native asset distributions, it collects the native coins into a `native_funds` vector.After the asset iteration loop, there is an unconditional `submsgs.push` that creates a `DistributeNative` submessage from `native_funds` and adds it to submsgs.This `DistributeNative` submessage will be added even if `native_funds` is empty. It is best practice to ensure that fund related messages are not empty before they are dispatched.

Recommendation

We recommend adding a check for `!native_funds.is_empty` before creating and pushing the `DistributeNative` submessage. This will prevent unnecessary empty distributed calls from being made.

43. Missing duplicate validation in `initialize_stakers`

RISK IMPACT: LOW	STATUS: RESOLVED

Description

In the `initialize_stakers` function in `packages/token-staking-impl/src/execute.rs:91`, if there is a duplicate member address in the `stakers` vector, the `user_stakes_sum` value will be larger than intended, causing discrepancies when summing up all members' weights.

Recommendation

We recommend deduping the `stakers` vector to ensure member addresses are unique.

44. Reduce gas usage using max limit instead of default limit in `get_versions_between_current_and_target`

RISK IMPACT: INFO	STATUS: RESOLVED

Description

The `get_versions_between_current_and_target` function in `contracts/enterprise/src/contract.rs:248` sets the `limit` as `None` when performing a `Versions` query. This means the returned versions will be 10 due to `DEFAULT_QUERY_LIMIT` in `contracts/enterprise-versioning/src/contract.rs:120`.

Recommendation

We recommend the `MAX_QUERY_LIMIT` which returns 50 `versions_response` to reduce overall query attempts.

45. Gas usage can be reduced in `import_cw3_membership` function

RISK IMPACT: INFO	STATUS: PARTIALLY RESOLVED

Revision Notes

The team has mentioned that they have chosen not to address the second part of the recommendation, which pertains to breaking early if they receive fewer items than the specified limit. This decision is based on the uncertainty of the limit that the CW3 contract might internally choose, which could potentially be lower than the limit provided by them.

Description

In the `import_cw3_membership` function in `contracts/enterprise-factory/src/multisig_membership.rs:34`, the pagination limit for voters is set to `None`. This will cause the list_votes function to unwrap the limit as DEFAULT_LIMIT, which is 10.

Since the intended usage for the `import_cw3_membership` function is to collect all voters from the CW3 contract, gas usage can be reduced by processing 50 members per query instead of 10 members per query. As dispatching queries costs gas, this approach reduces gas consumption by reducing the number of queries.

Other than that, the current implementation performs an extra query if all the voters are fetched. For example, if the first loop already fetches all the voters, the loop will continue due to line 52. After that, a query is dispatched to get all the voters in line 39. The loop will then finally break because there are no voters returned.

The extra loop can be removed by checking whether the returned voters are less than the provided `limit` value. If this is true, it means there are no voters remaining in the CW3 contract, and the loop can be exited early without performing an extra query. If the returned voters' length equals the provided limit,

it means that there are still remaining voters that need to be fetched, hence the loop needs to continue.

Recommendation

We recommend modifying the `limit` parameter to use `MAX_LIMIT` to reduce the number of queries dispatched. Additionally, the `last_voter` variable can be set to `None` after line `50` if the returned voters' length is less than the provided limit to reduce additional queries.

46. Event emitting wrong response

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `add_cross_chain_proxy` function in `contracts/enterprise/src/contract.rs:332`, the response being emitted calls the `execute_add_cross_chain_treasury_response` function when it should be calling the `execute_add_cross_chain_proxy_response` function instead.

Additionally, in the `execute_cast_council_vote_response` function in `packages/enterprise-governance-controller-api/src/response.rs:49`, the value of the "*action*" attribute key emitted "`cast_vote`" when it should be "`cast_council_vote`".

This is misleading because other emitted actions follow the executed function name.

Recommendation

We recommend modifying the values to the associated function name.

47. Gas usage can be reduced by breaking out of the loop early

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `get_versions_between_current_and_target` function in `contracts/enterprise/src/contract.rs:269`, the break in line `269` will not break the main loop in line `243`, the same applies for line `263`.

Instead, the loop continues iterating until it reaches line `253`. Especially when there are higher versions than the specified `target_version`, transaction failures may occur due to running out of gas.

Recommendation

We recommend modifying the function to break out of the main loop.

48. `add_cross_chain_proxy` function emits incorrect response

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `add_cross_chain_proxy` function in `contracts/enterprise/src/contract.rs:332`, the function should emit a response by calling `execute_add_cross_chain_proxy_response` instead of `execute_add_cross_chain_treasury_response`.

Recommendation

We recommend emitting the relevant attributes or events.

49. Incorrect label when instanting enterprise contract

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `create_enterprise_contract` function in `contracts/enterprise-treasury/src/migration.rs:209`, the init enterprise contract shows the label as *"Enterprise treasury"*, which is incorrect.

Recommendation

We recommend modifying the label to enterprise only.

50. Rename variable for clarity

RISK IMPACT: INFO	STATUS: RESOLVED

Description

The value of `TOTAL_WEIGHT` is the sum of all `EFFECTIVE_USER_WEIGHTS` and not of `USER_WEIGHTS`. This distinction is important to prevent potential misunderstandings.

Recommendation

We recommend renaming the `TOTAL_WEIGHT` to `EFFECTIVE_TOTAL_WEIGHT`.

51. `user_stake` is not emitted

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `unstake` function in `packages/denom-staking-impl/src/execute.rs:92-96`, the unstake event does not emit "*user_stake*" like the `stake_denom` function in `packages/denom-staking-impl/src/execute.rs:56`.

Recommendation

We recommend emitting the "*user_stake*" for consistency.

52. Council membership and attestation address are not emitted

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `execute_finalize_instantiation_response` function in `packages/enterprise-protocol/src/response.rs:7-27`, both the council membership and attestation are not emitted in the response.

Recommendation

We recommend emitting these values.

53. Query performed directly when associated function exists

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `update_gov_config` function in `contracts/enterprise-governance-controller/src/contract.rs:720-728`, `DaoInfoResponse` and `ComponentContractsResponse` are queried. However, instead of utilizing the corresponding functions designed for querying these responses, the queries are being executed directly.

Recommendation

We recommend using the corresponding query functions `query_dao_type` and `query_enterprise_components` for the respective queries.

54. Proposal UpdateCouncil can update only all fields

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Revision Notes

The team has mentioned that they plan to address this issue in a future revision, as it appears to require a disproportionately large amount of effort to address this specific matter in comparison to the benefits it would bring at this moment.

Description

In an `UpdateCouncil` type proposal, individual fields cannot be modified selectively. All fields must be specified, even those not intended for change. This could lead to unintentional modifications in some fields.

Recommendation

To address this issue, we recommend implementing a dedicated structure where each field is optional. By doing so, fields that are intended for modification can be explicitly set to `Some`, signaling a change. Conversely, fields that should remain unchanged can be designated as `None`. This approach provides clarity and reduces the risk of unintentional modifications.

55. ENTERPRISE_CODE_IDS is unused

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Revision Notes

The team has mentioned that the `ENTERPRISE_CODE_IDS` is necessary for older DAO versions to function properly.

Description

In `contracts/enterprise-factory/src/state.rs:21` of the `enterprise-factory` contract, the query for `ENTERPRISE_CODE_IDS` is exposed but unused.

Recommendation

We recommend removing the unused `ENTERPRISE_CODE_IDS` storage.

56. Proxy contract does not implement queries

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In `src/contract.rs:267-269` of the `proxy` contract, the `query` function exists without any queries implemented. External contracts and users are not able to retrieve information from the proxy contract due to the lack of these queries.

Recommendation

Consider exposing smart contract queries to return the values of `CONFIG` and `ACTIVE_REPLY_CALLBACKS`.

Document Control

Version	Date	Notes
-	24th August 2023	Security audit commencement date.
0.1	14th September 2023	Initial report with identified findings delivered.
0.5	22nd September 2023	Fixes remediations implemented and reviewed.
1.0	5th October 2023	Audit completed, final report delivered.

Appendices

A. Appendix – Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

Risk Level	Range
CRITICAL	10
SEVERE	From 9 to 8
MODERATE	From 7 to 6
LOW	From 5 to 4
INFORMATIONAL	From 3 to 1

LIKELIHOOD and **IMPACT** would be individually assessed based on the below:

Rate	LIKELIHOOD	IMPACT
5	**Extremely Likely**	Could result in severe and irreparable consequences.
4	**Likely**	May lead to substantial impact or loss.
3	**Possible**	Could cause partial impact or loss on a wide scale.
2	**Unlikely**	Might cause temporary disruptions or losses.
1	**Rare**	Could have minimal or negligible impact.

B. Appendix – Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended for discussion purposes only. The content of this audit report is provided "as is," without representations and warranties of any kind, and SCV-Security disclaims any liability for damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

THANK YOU FOR CHOOSING

SCV SECURITY

scv.services

contact@scv.services