A41 - SuperNova

Cosmos Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 4th, 2022 - October 31st, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Edits	10/15/2022	Gokberk Gulgun
0.2	Document Updates	10/25/2022	Gokberk Gulgun
0.3	Draft Review	10/31/2022	Gabi Urrutia
1.0	Remediation Plan	11/10/2022	Gokberk Gulgun
1.1	Remediation Plan Review	11/11/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Gokberk Gulgun	Halborn	Gokberk.Gulgun@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

A41 Team engaged Halborn to conduct a security audit on their **cosmos modules**, beginning on October 4th, 2022 and ending on October 31st, 2022 . The security assessment was scoped to the code base provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided nearly five weeks for the engagement and assigned two full-time security engineers to audit the security of the **modules**. The security engineers are blockchain and smart-contract security experts with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit to achieve the following:

- Ensure that SuperNOVA Module functionalities as intended.
- Identify potential security issues with the A41 Team.

In summary, Halborn identified few security risks that were mostly addressed by A41 Team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of the module. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of structures and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Static Analysis of security for scoped repository, and imported functions. (staticcheck, gosec, unconvert, LGTM, ineffassign and semgrep).
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.
- Dynamic Analysis on module functions and data types.

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating

a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE:

The security assessment was scoped to Carina-labs/nova/tree/v0.6.3 repository.

Commt ID - TREE

IN-SCOPE MODULES :

- airdrop
- gal
- icacontrol
- mint
- oracle
- poolincentive

FIX BRANCH & COMMIT ID:

- TREE
- Commit ID

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
1	3	6	13	7

EXECUTIVE OVERVIEW

LIKELIHOOD

IMPACT

		(HAL-04)		(HAL-01)
		(HAL-07)	(HAL-02) (HAL-03)	
(HAL-11) (HAL-12) (HAL-13) (HAL-14) (HAL-17) (HAL-18) (HAL-19) (HAL-20) (HAL-21) (HAL-23)		(HAL-05) (HAL-06) (HAL-08) (HAL-09) (HAL-10)		
	(HAL-15) (HAL-16) (HAL-22)			
(HAL-24) (HAL-25) (HAL-26) (HAL-27) (HAL-28) (HAL-29) (HAL-30)				

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - FUNDS CAN BE LOCKED IF THE ZONE DECIMAL IS HIGHER THAN 18	Critical	SOLVED - 11/10/2022
HAL-02 - UNIQUENESS OF ZONES ARE NOT VALIDATED	High	SOLVED - 11/10/2022
HAL-03 - MISSING VALIDATION LEADS TO LOST OF FUNDS	High	SOLVED - 11/10/2022
HAL-04- NON-DETERMINISTIC ITERATIONS CAN CAUSE CONSENSUS FAILURES	High	SOLVED - 11/10/2022
HAL-05 - BLOCK HEIGHT IS NOT CHECKED WHEN UPDATING STATE	Medium	SOLVED - 11/10/2022
HAL-06 - DUPLICATED ZONE LIST IS NOT REMOVED DURING THE GENESIS INITIALIZATION	Medium	RISK ACCEPTED
HAL-07 - MISSING VALIDATION ON THE HOST DENOM AND IBC DENOM	Medium	SOLVED - 11/10/2022
HAL-08 - MISSING FUNCTIONALITY WHEN CONNECTION CHANNEL IS CLOSED	Medium	RISK ACCEPTED
HAL-09 - ORACLE STATE IS GENERATED INSTEAD OF UPDATE	Medium	SOLVED - 11/10/2022
HAL-10 - NON-DETERMINISTIC SYSTEM TIME	Medium	SOLVED - 11/10/2022
HAL-11 - IBC TIMEOUT IS HARDCODED	Low	RISK ACCEPTED
HAL-12 - CODECS DO NOT HAVE INIT FUNCTION	Low	SOLVED - 11/10/2022
HAL-13 - CENTRALIZATION RISK	Low	SOLVED - 11/10/2022
HAL-14 - DELEGATE MESSAGES WITH ZERO AMOUNT IS NOT CHECKED	Low	SOLVED - 11/10/2022
HAL-15 - LACK OF SIMULATION AND FUZZING OF THE MODULE INVARIANT	Low	RISK ACCEPTED
HAL-16 - LACK OF EVENT EMISSION IS BAD PRACTICE	Low	SOLVED - 11/10/2022

HAL-17 - LACK OF SPEC ON THE MODULES	Low	SOLVED - 11/10/2022
HAL-18 - COMPUTATIONALLY HEAVY OPERATIONS IN BEGINBLOCKER MAY SLOW DOWN OR STOP BLOCK PRODUCTION	Low	SOLVED - 11/10/2022
HAL-19 - TEST DOCKER IMAGE RUNNING AS ROOT	Low	RISK ACCEPTED
HAL-20 - CSWASM CONTRACT ADDRESS IS NOT VALIDATED ON THE GENESIS STATE	Low	SOLVED - 11/10/2022
HAL-21 - LACK OF VERSION RECORD EXISTENCE CHECK	Low	SOLVED - 11/10/2022
HAL-22 - CONFLICT BETWEEN UNDELEGATE AND TRANSFER FAIL HOOKS	Low	SOLVED - 11/10/2022
HAL-23 - MISSING UPPER BOUND DEFINITION ON THE MAX ENTRIES	Low	RISK ACCEPTED
HAL-24 - MODULES DO NOT USE REST CLI HANDLER	Informational	ACKNOWLEDGED
HAL-25 - OPEN TODOs	Informational	ACKNOWLEDGED
HAL-26 - PANIC IS USED FOR ERROR HANDLING	Informational	SOLVED - 11/10/2022
HAL-27 - TYPO ON THE MODULE NAME	Informational	SOLVED - 11/10/2022
HAL-28 - USE OF DEBUG LOGGER INSTEAD OF ERROR	Informational	ACKNOWLEDGED
HAL-29 - LACK OF EXTENSIVE TEST COVERAGE	Informational	SOLVED - 11/10/2022
HAL-30 - UNUSED CODE NEGATIVELY IMPACTS MAINTAINABILITY	Informational	SOLVED - 11/10/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) FUNDS CAN BE LOCKED IF THE ZONE DECIMAL IS HIGHER THAN 18 - CRITICAL

Description:

During the code review, It has been noticed that zone decimal does not have upper bound. Zone decimal can be added with more than 18 decimals. On the **ClaimSnMesssage**, ClaimShareToken is used when user want to claim their share token. However, If the zone decimal is higher than 18, the following equation will fail and assets could not claim from the system. The oracle address directly can break system through decimal.

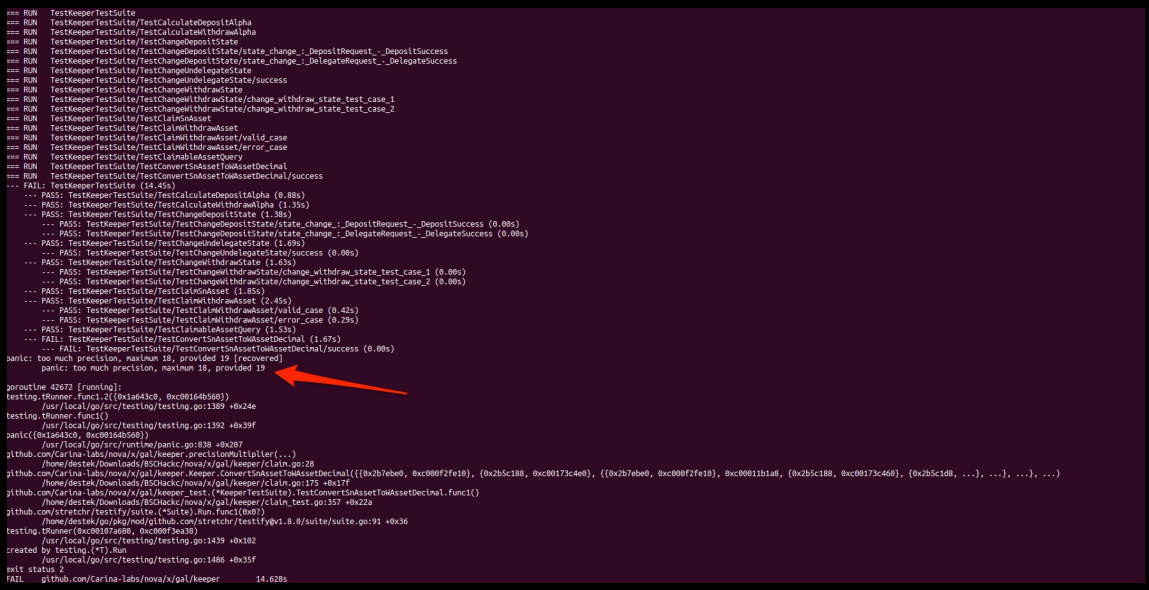
Code Location:

/x/gal/keeper/claim.go, Lines 167

Listing 1

```
1 func (k Keeper) ConvertWAssetToSnAssetDecimal(amount *big.Int,
↳ decimal int64, denom string) sdk.Coin {
2     convertDecimal := snAssetDecimal - decimal
3     asset := new(big.Int).Mul(amount, precisionMultiplier(0))
4     snAsset := new(big.Int).Quo(asset, precisionMultiplier(
↳ convertDecimal))
5     return sdk.NewCoin(denom, sdk.NewIntFromBigInt(snAsset))
6 }
7
```

Proof Of Concept:



Scenario:

- Register/Change zone with more than 18 decimals.
- Even if It's privileged function by the controller, GAL module will lead to chain halt due to above equation.

Listing 2

```
var (
        transferPort    = "transfer"
        transferChannel = "channel-0"
        icaConnection   = "connection-1"

        zoneId        = "baseZone"
        baseOwnerAcc  = sdk.AccAddress(secp256k1.GenPrivKey().
↳ PubKey().Address())
        baseDenom     = "stake"
        baseSnDenom   = "snstake"
        baseDecimal   = int64(19)
)
```

```
zones:
- base_denom: uatom
  decimal: "6"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: cosmos1q5hr70zy095weyygjpys2rak7zeda9lu78t9k08mwe5a2sdwn7yqkktx5f
  ica_connection_info:
    connection_id: connection-4
    port_id: gaia.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snuatom
  transfer_info:
    channel_id: channel-2
    port_id: transfer
  validator_address: cosmosvaloper1zkarsurgym3hnm06qupyt96pu0k24k4fq9la7n
  zone_id: gaia
- base_denom: ujuno
  decimal: "6"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: juno173jm2gjfznv5hrkkl80nkl2a6e7u52r5jyytjyvp6vgjk9d8x8jq765y7e
  ica_connection_info:
    connection_id: connection-2
    port_id: juno.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snujuno
  transfer_info:
    channel_id: channel-1
    port_id: transfer
  validator_address: junovaloper1zkarsurgym3hnm06qupyt96pu0k24k4fv77uw9
  zone_id: juno
- base_denom: uosmo
  decimal: "256"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: osmo18nvvdqysc0nqkkzqyfppnxfmn0hu4k7557tdg4j73znpev0p99fqscchh4
  ica_connection_info:
    connection_id: connection-0
    port_id: osmosis.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snuosmo
  transfer_info:
    channel_id: channel-0
    port_id: transfer
  validator_address: osmovaloper1zkarsurgym3hnm06qupyt96pu0k24k4fhasmn4
  zone_id: osmosis
```

Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

On the **ICAControl** module, ensure that zone decimal is not higher than **18**.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 08edd624 by adding the decimal check.

3.2 (HAL-02) UNIQUENESS OF ZONES ARE NOT VALIDATED - HIGH

Description:

Uniqueness of denom is checked with zone ID on the Registered zones. **BaseDenom** should be unique for each zone. Even if Its controller privileged function. One wrong value set on the denom will be resulted with the funds lost. We recommend ensuring that both **BaseDenom** is unique throughout all host zones before allowing them to be set in the **RegisterZone**.

Code Location:

/x/icacontrol/keeper/msg_server.go, Lines 99

Listing 3

```
1  func (k msgServer) ChangeRegisteredZone(goCtx context.Context,
↳ zone *types.MsgChangeRegisteredZone) (*types.
↳ MsgChangeRegisteredZoneResponse, error) {
2      ctx := sdk.UnwrapSDKContext(goCtx)
3
4      if !k.IsValidControllerAddr(ctx, zone.ZoneId, zone.IcaAccount.
↳ ControllerAddress) {
5          return nil, sdkerrors.Wrap(sdkerrors.ErrInvalidAddress,
↳ zone.IcaAccount.ControllerAddress)
6      }
7
8      zoneInfo := &types.RegisteredZone{
9          ZoneId: zone.ZoneId,
10         IcaConnectionInfo: &types.IcaConnectionInfo{
11             ConnectionId: zone.IcaInfo.ConnectionId,
12             PortId:       zone.IcaInfo.PortId,
13         },
14         IcaAccount: &types.IcaAccount{
15             ControllerAddress: zone.IcaAccount.ControllerAddress,
16             HostAddress:       zone.IcaAccount.HostAddress,
17         },
18         TransferInfo: &types.TransferConnectionInfo{
```

```
19              PortId:    zone.TransferInfo.PortId,
20              ChannelId: zone.TransferInfo.ChannelId,
21          },
22          ValidatorAddress: zone.ValidatorAddress,
23          BaseDenom:        zone.BaseDenom,
24          SnDenom:          appendSnPrefix(types.PrefixSnAsset, zone
↳ .BaseDenom),
25          Decimal:          zone.Decimal,
26          MaxEntries:       zone.MaxEntries,
27      }
28
29      k.Keeper.RegisterZone(ctx, zoneInfo)
30      return &types.MsgChangeRegisteredZoneResponse{
31          ZoneId:           zoneInfo.ZoneId,
32          IcaInfo:          zoneInfo.IcaConnectionInfo,
33          TransferInfo:     zoneInfo.TransferInfo,
34          ValidatorAddress: zoneInfo.ValidatorAddress,
35          BaseDenom:        zoneInfo.BaseDenom,
36          SnDenom:          zoneInfo.BaseDenom,
37          Decimal:          zoneInfo.Decimal,
38          MaxEntries:       zoneInfo.MaxEntries,
39      }, nil
40  }
41
```

Proof Of Concept:

```
                                        $ ./build/novad query icacontrol all-zone
zones:
- base_denom: uatom
  decimal: "6"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: cosmos1q5hr70zy095weyygjpys2rak7zeda9lu78t9k08mwe5a2sdwn7yqkktx5f
  ica_connection_info:
    connection_id: connection-4
    port_id: gaia.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snuatom
  transfer_info:
    channel_id: channel-2
    port_id: transfer
  validator_address: cosmosvaloper1zkarsurgym3hnm06qupyt96pu0k24k4fq9la7n
  zone_id: gaia
- base_denom: ujuno
  decimal: "6"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: juno173jm2gjfznv5hrkkl80nkl2a6e7u52r5jyytjyvp6vgjk9d8x8jq765y7e
  ica_connection_info:
    connection_id: connection-2
    port_id: juno.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snujuno
  transfer_info:
    channel_id: channel-1
    port_id: transfer
  validator_address: junovaloper1zkarsurgym3hnm06qupyt96pu0k24k4fv77uw9
  zone_id: juno
- base_denom: ujuno
  decimal: "6"
  ica_account:
    controller_address: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
    host_address: juno173jm2gjfznv5hrkkl80nkl2a6e7u52r5jyytjyvp6vgjk9d8x8jq765y7e
  ica_connection_info:
    connection_id: connection-0
    port_id: osmosis.nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
  max_entries: "100"
  sn_denom: snujuno
  transfer_info:
    channel_id: channel-0
    port_id: transfer
  validator_address: junovaloper1zkarsurgym3hnm06qupyt96pu0k24k4fv77uw9
  zone_id: osmosis
```

Risk Level:

Likelihood - 4
Impact - 4

Recommendation:

Ensure that all zones are unique and compared with existing zones.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit b6041991 by adding the validation in the zones.

3.3 (HAL-03) MISSING VALIDATION LEADS TO LOST OF FUNDS - HIGH

Description:

On the minting module, at the beginning of block, every time tokens are minted depends on the staking amount. However, If the pool does not exist or contract address is typed wrongly. Minting module can panic, and the chain can halt. On the **PoolIncentive** module, Pool Contract address is not verified If the distribution operation fails than directly chain will halt with this check.

Code Location:

/x/mint/keeper/keeper.go, Lines 175

Listing 4

```
func (k Keeper) distributeLPIncentivePools(ctx sdk.Context, denom string) error {
    pools := k.PoolIncentiveKeeper.GetAllIncentivePool(ctx)
    if len(pools) == 0 {
        return nil
    }

    totalWeight := k.PoolIncentiveKeeper.GetTotalWeight(ctx)
    moduleAddr := k.accountKeeper.GetModuleAddress(types.LpIncentiveModuleAccName)
    lpIncentiveCoin := k.bankKeeper.GetBalance(ctx, moduleAddr, denom)

    for _, pool := range pools {
        poolWeight := sdk.NewIntFromUint64(pool.Weight).ToDec().Quo(sdk.NewIntFromUint64(totalWeight).ToDec())
        incentive := sdk.NewDecFromInt(lpIncentiveCoin.Amount).Mul(poolWeight)
        incentivesCoins := sdk.NewCoins(sdk.NewCoin(lpIncentiveCoin.Denom, incentive.TruncateInt()))
        poolAddr, err := sdk.AccAddressFromBech32(pool.PoolContractAddress)
```

```
16          if err != nil {
17              return err
18          }
19
20          err = k.bankKeeper.SendCoinsFromModuleToAccount(ctx, types
↳ .LpIncentiveModuleAccName, poolAddr, incentivesCoins)
21          if err != nil {
22              return err
23          }
24      }
25      return nil
26  }
```

Proof Of Concept:

Listing 5

```
 1 func (suite *KeeperTestSuite) TestCreateIncentivePool() {
 2          tcs := []struct {
 3                  name      string
 4                  preset    []types.IncentivePool
 5                  pool      types.IncentivePool
 6                  shouldErr bool
 7          }{
 8                  {
 9                          name:   "valid case",
10                          preset: []types.IncentivePool{},
11                          pool: types.IncentivePool{
12                                  PoolId:              "
↳ poolincentive-1",
13                                  PoolContractAddress: "12345",
14                                  Weight:              0,
15                          },
16                          shouldErr: false,
17                  },
18          }
19          keeper := suite.App.PoolKeeper
20
21          for _, tc := range tcs {
22                  suite.Run(tc.name, func() {
23                          // setup
24                          for i := range tc.preset {
```

```
25                                      err := keeper.CreateIncentivePool(
↳ suite.Ctx, &tc.preset[i])
26                                      suite.NoError(err)
27                              }
28
29                              err := keeper.CreateIncentivePool(suite.
↳ Ctx, &tc.pool)
30                              if tc.shouldErr {
31                                      suite.Error(err)
32                              } else {
33                                      suite.NoError(err)
34
35                                      incentivePool, err := keeper.
↳ FindIncentivePoolById(suite.Ctx, tc.pool.PoolId)
36                                      suite.NoError(err)
37                                      suite.Equal(tc.pool.PoolId,
↳ incentivePool.PoolId)
38                                      suite.Equal(tc.pool.
↳ PoolContractAddress, incentivePool.PoolContractAddress)
39                                      suite.Equal(tc.pool.Weight,
↳ incentivePool.Weight)
40                              }
41                      })
42              }
43 }
```

Risk Level:

Likelihood - 4
Impact - 4

Recommendation:

Ensure that pool address is validated in the **ValidateBasic** function. **ValidateBasic** is happening during the **CheckTx** phase, and it doesn't have access to the state. In the current implementation, only signer is validated on the **ValidateBasic** function.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 538abc771 by adding the pool contract address validation.

3.4 (HAL-04) NON-DETERMINISTIC ITERATIONS CAN CAUSE CONSENSUS FAILURES - HIGH

Description:

In several instances of the codebase, iterations are done over maps. Since Go map iterations are non-deterministic, this would cause each validator to produce a different app hash, causing a consensus failure and potentially leading to a chain halt.

Code Location:

Listing 6

```
x/gal/keeper/grpc_query.go:111
x/gal/keeper/withdraw.go:78
x/gal/keeper/withdraw.go:135
x/gal/keeper/withdraw.go:156
x/gal/keeper/withdraw.go:170
x/gal/keeper/withdraw.go:186
x/gal/keeper/withdraw.go:221
app/app.go:373
app/app.go:451
x/airdrop/keeper/user_state.go:71
```

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

We recommend sorting the map keys into a slice and iterating over the sorted keys to ensure deterministic results among all validators.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 0584f524 by adding stable_marshaler into the protobuf. With this option, **Marshaler** the output is guaranteed to be deterministic.

3.5 (HAL-05) BLOCK HEIGHT IS NOT CHECKED WHEN UPDATING STATE - MEDIUM

Description:

The oracle module manages the status of the zones associated with the Supernova protocol. The status includes the amount of coins delegated to the Zone's Validator, block height, proof, and so on. This information is injected by the bot at a short interval. For the integrity of the information, the **AppHash** and **block height** are provided together. During the code review, It has been observed block height is not checked when updating oracle state.

Code Location:

/x/oracle/keeper/msg_server.go, Lines 32

Listing 7

```
 1 func (server msgServer) UpdateChainState(goctx context.Context,
↳ state *types.MsgUpdateChainState) (*types.
↳ MsgUpdateChainStateResponse, error) {
 2     ctx := sdk.UnwrapSDKContext(goctx)
 3     if !server.keeper.IsValidOracleAddress(ctx, state.ZoneId,
↳ state.Operator) {
 4         return nil, types.ErrInvalidOperator
 5     }
 6
 7     newOracleState := &types.ChainInfo{
 8         Coin:            state.Coin,
 9         OperatorAddress: state.Operator,
10         LastBlockHeight: state.BlockHeight,
11         AppHash:         state.AppHash,
12         ZoneId:          state.ZoneId,
13     }
14
15     oracleVersion, _ := server.keeper.GetOracleVersion(ctx, state.
↳ ZoneId)
```

```
16
17      trace := types.IBCTrace{
18          Version: oracleVersion + 1,
19          Height:  uint64(ctx.BlockHeight()),
20      }
21      server.keeper.SetOracleVersion(ctx, state.ZoneId, trace)
22
23      if err := server.keeper.UpdateChainState(ctx, newOracleState);
↳  err != nil {
24          return nil, sdkerrors.Wrapf(types.ErrUnknown, "err: %v",
↳ err)
25      }
26
27      if err := ctx.EventManager().EmitTypedEvent(newOracleState);
↳ err != nil {
28          return nil, sdkerrors.Wrapf(types.ErrUnknown, "err: %v",
↳ err)
29      }
30
31      return &types.MsgUpdateChainStateResponse{}, nil
32 }
```

Proof Of Concept:

```
=== RUN   TestKeeperTestSuite
=== RUN   TestKeeperTestSuite/TestExportGenesis
=== RUN   TestKeeperTestSuite/TestGRPCState
=== RUN   TestKeeperTestSuite/TestGRPCState/should_get_state
=== RUN   TestKeeperTestSuite/TestGRPCState/should_get_error
=== RUN   TestKeeperTestSuite/TestQueryOracleVersion
=== RUN   TestKeeperTestSuite/TestQueryParam
=== RUN   TestKeeperTestSuite/TestQueryParam/empty_operator
=== RUN   TestKeeperTestSuite/TestQueryParam/got_operator
=== RUN   TestKeeperTestSuite/TestRegisterOracleAddress
=== RUN   TestKeeperTestSuite/TestServerUpdateChainState
LAST BLOCK HEIGHT : 99999999=== RUN   TestKeeperTestSuite/TestSetOracleVersion
=== RUN   TestKeeperTestSuite/TestUpdateChainState
=== RUN   TestKeeperTestSuite/TestUpdateChainState/no_operator
=== RUN   TestKeeperTestSuite/TestUpdateChainState/no_data_with_incorrect_query
=== RUN   TestKeeperTestSuite/TestUpdateChainState/should_success
--- PASS: TestKeeperTestSuite (0.56s)
    --- PASS: TestKeeperTestSuite/TestExportGenesis (0.14s)
    --- PASS: TestKeeperTestSuite/TestGRPCState (0.09s)
        --- PASS: TestKeeperTestSuite/TestGRPCState/should_get_state (0.00s)
        --- PASS: TestKeeperTestSuite/TestGRPCState/should_get_error (0.00s)
    --- PASS: TestKeeperTestSuite/TestQueryOracleVersion (0.06s)
    --- PASS: TestKeeperTestSuite/TestQueryParam (0.07s)
        --- PASS: TestKeeperTestSuite/TestQueryParam/empty_operator (0.00s)
        --- PASS: TestKeeperTestSuite/TestQueryParam/got_operator (0.00s)
    --- PASS: TestKeeperTestSuite/TestRegisterOracleAddress (0.05s)
    --- PASS: TestKeeperTestSuite/TestServerUpdateChainState (0.05s)
    --- PASS: TestKeeperTestSuite/TestSetOracleVersion (0.05s)
    --- PASS: TestKeeperTestSuite/TestUpdateChainState (0.05s)
        --- PASS: TestKeeperTestSuite/TestUpdateChainState/no_operator (0.00s)
        --- PASS: TestKeeperTestSuite/TestUpdateChainState/no_data_with_incorrect_query (0.00s)
        --- PASS: TestKeeperTestSuite/TestUpdateChainState/should_success (0.00s)
PASS
ok      github.com/Carina-labs/nova/x/oracle/keeper     0.648s
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

Consider using latest **Block height** from the context.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit e938023 by adding the block height check.

3.6 (HAL-06) DUPLICATED ZONE LIST IS NOT REMOVED DURING THE GENESIS INITIALIZATION - MEDIUM

Description:

In /x/icacontrol/keeper/genesis.go, the **icacontrol** genesis validate functionality does not remove duplicates from the zone list slice. As the **RegisterZone** keeper function in /x/icacontrol/keeper/zone.go#L15 uses the zone ID as the key identifier, having duplicate chain ID values in the **RegisterZone** slice would cause the final index with the same chain ID value to be stored in the storage. As a result, previous zone configurations with duplicate zone ID values would be overwritten and ignored completely.

Code Location:

/x/icacontrol/keeper/msg_server.go, Lines 99

Listing 8

```
1 func (k Keeper) InitGenesis(ctx sdk.Context, genState *types.
↳ GenesisState) {
2     k.SetParams(ctx, genState.Params)
3
4     for _, controllerInfo := range genState.ControllerAddressInfo
↳ {
5         k.SetControllerAddr(ctx, controllerInfo.ZoneId,
↳ controllerInfo.ControllerAddress)
6     }
7 }
8
9 func (k Keeper) ExportGenesis(ctx sdk.Context) *types.GenesisState
↳  {
10     params := k.GetParams(ctx)
11     return types.NewGenesisState(params)
12 }
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

Add need validation mechanisms in genesis.

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of issue. They will fix it in a future release.

3.7 (HAL-07) MISSING VALIDATION ON THE HOST DENOM AND IBC DENOM - MEDIUM

Description:

ValidateDenom is the default validation function for **Host.Denom**. In the parameters, Denom is only checked with the length. The system should verify all parameters, even if the controller manages them.

Code Location:

/x/icacontrol/types/msgs.go, Lines 50

Listing 9

```
 1 func (msg MsgRegisterZone) ValidateBasic() error {
 2     if strings.TrimSpace(msg.ZoneId) == "" {
 3         return sdkerrors.Wrapf(ErrZoneIdNotNil, "zoneId is not nil
↳ ")
 4     }
 5
 6     if strings.TrimSpace(msg.IcaInfo.ConnectionId) == "" {
 7         return errors.New("missing ICA connection ID")
 8     }
 9
10     _, err := sdk.AccAddressFromBech32(msg.IcaAccount.
↳ ControllerAddress)
11     if err != nil {
12         return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "
↳ Invalid controller address")
13     }
14
15     if strings.TrimSpace(msg.ValidatorAddress) == "" {
16         return errors.New("missing validator address")
17     }
18
19     if strings.TrimSpace(msg.BaseDenom) == "" {
20         return errors.New("missing denom")
```

```
21      }
22
23      if msg.MaxEntries == 0 {
24          return errors.New("cannot set max_entries to zero")
25      }
26
27      return nil
28  }
29
```

Proof Of Concept:

Listing 10

```
 1  package keeper_test
 2
 3  import (
 4          novatesting "github.com/Carina-labs/nova/testing"
 5          icacontroltypes "github.com/Carina-labs/nova/x/icacontrol/
↳  types"
 6          "github.com/cosmos/cosmos-sdk/crypto/keys/secp256k1"
 7          sdk "github.com/cosmos/cosmos-sdk/types"
 8          icatypes "github.com/cosmos/ibc-go/v3/modules/apps/27-
↳  interchain-accounts/types"
 9          ibcchanneltypes "github.com/cosmos/ibc-go/v3/modules/core
↳  /04-channel/types"
10  )
11
12  var (
13          transferPort    = "transfer"
14          transferChannel = "channel-0"
15          icaConnection   = "connection-1"
16
17          zoneId       = "baseZone"
18          baseOwnerAcc = sdk.AccAddress(secp256k1.GenPrivKey().
↳  PubKey().Address())
19          baseDenom    = "NOVA"
20          baseSnDenom  = "snNOVA"
21          baseDecimal  = int64(19)
22  )
```

Risk Level:

Likelihood - 3
Impact - 4

Recommendation:

It is recommended to use ValidateDenom with ValidateIBCDenom in the related sections. On the other hand, it is recommended to verify the connection through **IBC**.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 284db08a by adding validations.

3.8 (HAL-08) MISSING FUNCTIONALITY WHEN CONNECTION IS CLOSED - MEDIUM

Description:

When the IBC channel connection was closed, the GAL module was unable to handle the workflow over the connection.

Code Location:

/x/icacontrol/ibc_mobule.go, Lines 109

Listing 11

```
1 func (im IBCModule) OnChanCloseConfirm(
2     ctx sdk.Context,
3     portID,
4     channelID string,
5 ) error {
6     return nil
7 }
8
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

It is recommended to handle the connection when it is closed and confirmed.

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of this issue. They will fix it in a future release.

3.9 (HAL-09) ORACLE STATE IS GENERATED INSTEAD OF UPDATE - MEDIUM

Description:

The status of the zones connected to the Supernova protocol is managed by the Oracle module. The status contains information such as block height, proof, block size, and the number of coins assigned to the zone validator. However, when the state is updated, a new state is added instead of an update. The implementation does not check the uniqueness of the zone ID.

Code Location:

/x/oracle/keeper/msg_server.go#L23

Listing 12

```
func (server msgServer) UpdateChainState(goctx context.Context,
↳ state *types.MsgUpdateChainState) (*types.
↳ MsgUpdateChainStateResponse, error) {
    ctx := sdk.UnwrapSDKContext(goctx)
    if !server.keeper.IsValidOracleAddress(ctx, state.ZoneId,
↳ state.Operator) {
        return nil, types.ErrInvalidOperator
    }

    newOracleState := &types.ChainInfo{
        Coin:            state.Coin,
        OperatorAddress: state.Operator,
        LastBlockHeight: state.BlockHeight,
        AppHash:         state.AppHash,
        ZoneId:          state.ZoneId,
    }

    oracleVersion, _ := server.keeper.GetOracleVersion(ctx, state.
↳ ZoneId)

    trace := types.IBCTrace{
```

```
18          Version: oracleVersion + 1,
19          Height:  uint64(ctx.BlockHeight()),
20      }
21      server.keeper.SetOracleVersion(ctx, state.ZoneId, trace)
22
23      if err := server.keeper.UpdateChainState(ctx, newOracleState);
↳  err != nil {
24          return nil, sdkerrors.Wrapf(types.ErrUnknown, "err: %v",
↳ err)
25      }
26
27      if err := ctx.EventManager().EmitTypedEvent(newOracleState);
↳ err != nil {
28          return nil, sdkerrors.Wrapf(types.ErrUnknown, "err: %v",
↳ err)
29      }
30
31      return &types.MsgUpdateChainStateResponse{}, nil
32 }
```

Risk Level:

Likelihood - 3

Impact - 3

Proof Of Concept:

```
$ ./build/novad query oracle state uosmo
app_hash: dGVzdF9hcHBoYXNo
coin:
  amount: "1300000000"
  denom: uosmo
last_block_height: "100"
operator: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
zone_id: osmosis
$ ./build/novad query oracle state unova
app_hash: ZEdWemRGOWhjSEJvWVhObw==
coin:
  amount: "1300000000"
  denom: unova
last_block_height: "100"
operator: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
zone_id: juno
$ ./build/novad query oracle state ujuno
app_hash: dGVzdF9hcHBoYXNo
coin:
  amount: "1300000000"
  denom: ujuno
last_block_height: "100"
operator: nova1lds58drg8lvnaprcue2sqgfvjnz5ljlkq9lsyf
zone_id: juno
```

Recommendation:

Consider updating the state instead of generating a new state.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit bca885ebda by adding zone validation.

3.10 (HAL-10) NON-DETERMINISTIC SYSTEM TIME - MEDIUM

Description:

Using time.Now() returns the timestamp of the operating system. Local clock times are subjective and therefore non-deterministic. As there may be small discrepancies between the timestamp of various nodes, the chain may not be able to reach consensus. Cosmos SDK had a vulnerability named **jackfruit** which may with a consensus halt on the **x/authz** module. In the ICAControl module, the transaction send function uses **time.Now()** on the timestamp.

Code Location:

/x/icacontrol/keeper/send_msgs.go#L42

Listing 13

```
func (k Keeper) SendTx(ctx sdk.Context, controllerId, connectionId
↳  string, msgs []sdk.Msg) error {
    portID, err := icatypes.NewControllerPortID(controllerId)
    if err != nil {
        return err
    }

    channelID, found := k.IcaControllerKeeper.GetActiveChannelID(
↳ ctx, connectionId, portID)
    if !found {
        return sdkerrors.Wrapf(icatypes.ErrActiveChannelNotFound,
↳ "failed to retrieve active channel for port %s", portID)
    }

    chanCap, found := k.scopedKeeper.GetCapability(ctx, host.
↳ ChannelCapabilityPath(portID, channelID))
    if !found {
        return sdkerrors.Wrap(channeltypes.
↳ ErrChannelCapabilityNotFound, "module does not own channel
↳ capability")
    }
```

```
16
17      data, err := icatypes.SerializeCosmosTx(k.cdc, msgs)
18      if err != nil {
19          return err
20      }
21
22      packetData := icatypes.InterchainAccountPacketData{
23          Type: icatypes.EXECUTE_TX,
24          Data: data,
25      }
26
27      // timeoutTimestamp set to max value with the unsigned bit
↳ shifted to satisfy hermes timestamp conversion
28      // it is the responsibility of the auth module developer to
↳ ensure an appropriate timeout timestamp
29      timeoutTimestamp := time.Now().Add(time.Minute * 10).UnixNano
↳ ()
30      _, err = k.IcaControllerKeeper.SendTx(ctx, chanCap,
↳ connectionId, portID, packetData, uint64(timeoutTimestamp))
31      if err != nil {
32          return err
33      }
34
35      return nil
36  }
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

Consider using block time instead of **timestamp.Now()**.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 4c4bdac0 by changing the **timestamp.Now()** function with block time.

3.11 (HAL-11) IBC TIMEOUT IS HARDCODED - LOW

Description:

During code review, it has been noted that the timeout timestamp is hardcoded. From that reason, if the IBC connection exceeds the predefined timeout, it will fail.

Code Location:

/x/gal/keeper/ibc_transfer.go, Lines 39

Listing 14

```
func (k Keeper) TransferToTargetZone(ctx sdk.Context, option *
↳ IBCTransferOption) error {
    goCtx := sdk.WrapSDKContext(ctx)
    sender, err := sdk.AccAddressFromBech32(option.Sender)
    if err != nil {
        return err
    }

    _, err = k.ibcTransferKeeper.Transfer(goCtx,
        &transfertypes.MsgTransfer{
            SourcePort:    option.SourcePort,
            SourceChannel: option.SourceChannel,
            Token:         option.Token,
            Sender:        sender.String(),
            Receiver:      option.Receiver,
            TimeoutHeight: ibcclienttypes.Height{
                RevisionHeight: 0,
                RevisionNumber: 0,
            },
            TimeoutTimestamp: uint64(ctx.BlockTime().UnixNano() +
↳ 5*time.Minute.Nanoseconds()),
        },
    )

    return err
```

```
24 }
25
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Consider defining the timeout variable as a parameter.

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of this finding. They will fix it by taking the timeout as a cli variable.

3.12 (HAL-12) CODECS DO NOT HAVE INIT FUNCTION - LOW

Description:

The codec **init** function registers all Amino interfaces and concrete types in the module's amino codec. In the code base, **init** function is not implemented.

Code Location:

/x/gal/types/codec.go#L15

Listing 15

```
1 func RegisterLegacyAminoCodec(cdc *codec.LegacyAmino) {
2 }
3
4 func RegisterInterfaces(registry types.InterfaceRegistry) {
5     msgservice.RegisterMsgServiceDesc(registry, &_Msg_serviceDesc)
6     registry.RegisterImplementations(
7         (*sdk.Msg)(nil),
8     )
9 }
```

/x/icacontrol/types/codec.go

Listing 16

```
1 func RegisterCodec(cdc *codec.LegacyAmino) {
2     cdc.RegisterConcrete(MsgRegisterZone{}, "icacontrol/
↳ MsgRegisterZone", nil)
3     cdc.RegisterConcrete(MsgIcaDelegate{}, "icacontrol/
↳ MsgIcaDelegate", nil)
4     cdc.RegisterConcrete(MsgIcaUndelegate{}, "icacontrol/
↳ MsgIcaUndelegate", nil)
5     cdc.RegisterConcrete(MsgIcaTransfer{}, "icacontrol/
↳ MsgIcaTransfer", nil)
```

```
 6     cdc.RegisterConcrete(MsgIcaAutoStaking{}, "icacontrol/
↳ MsgIcaAutoStaking", nil)
 7 }
 8
 9 func RegisterInterfaces(registry cdctypes.InterfaceRegistry) {
10     msgservice.RegisterMsgServiceDesc(registry, &_Msg_serviceDesc)
11     registry.RegisterImplementations(
12         (*sdk.Msg)(nil),
13         &MsgRegisterZone{},
14         &MsgIcaDelegate{},
15         &MsgIcaUndelegate{},
16         &MsgIcaTransfer{},
17         &MsgIcaAutoStaking{},
18     )
19 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Ideally, Amino types should be registered inside this codec within the **init** function of each module's **codec.go**.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 709c77b by registering codecs.

3.13 (HAL-13) CENTRALIZATION RISK - LOW

Description:

During the withdrawing of funds, the address of the controller is used. It poses risk of centralization. If the private key is stolen, the funds can be stolen. Any compromise to the controller account can allow the hacker to manipulate the project through several functions.

Code Location:

/x/gal/keeper/withdraw.go#L197

Listing 17

```
1 func (k Keeper) ClaimWithdrawAsset(ctx sdk.Context, from sdk.
↳ AccAddress, withdrawer sdk.AccAddress, amt sdk.Coin) error {
2     err := k.bankKeeper.SendCoins(ctx, from, withdrawer, sdk.
↳ NewCoins(amt))
3     if err != nil {
4         return err
5     }
6
7     return nil
8 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended that the client to carefully manage the private key of the controller account to avoid any potential hacking risk. In general, it is strongly recommended enhancing centralized privileges or roles in

the protocol through a decentralized mechanism or module-based accounts with enhanced security practices.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit db0d7585 by using the module.

3.14 (HAL-14) DELEGATE MESSAGES WITH ZERO AMOUNT IS NOT CHECKED - LOW

Description:

DelegateMsgs function provides the list of **Delegate Txs** to execute based on the current state and parameters. However, the amount is not checked if it is greater than zero. During the execution of the IBC message, zero amount of delegations can fail in the system.

Code Location:

/x/gal/keeper/msg_server.go#L120

Listing 18

```
func (m msgServer) Delegate(goCtx context.Context, delegate *types
↳ .MsgDelegate) (*types.MsgDelegateResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    if !m.keeper.icaControlKeeper.IsValidControllerAddr(ctx,
↳ delegate.ZoneId, delegate.ControllerAddress) {
        return nil, sdkerrors.Wrap(sdkerrors.ErrInvalidAddress,
↳ delegate.ControllerAddress)
    }

    zoneInfo, ok := m.keeper.icaControlKeeper.GetRegisteredZone(
↳ ctx, delegate.ZoneId)
    if !ok {
        return nil, types.ErrNotFoundZoneInfo
    }

    // version state check
    if !m.keeper.IsValidDelegateVersion(ctx, delegate.ZoneId,
↳ delegate.Version) {
        return nil, sdkerrors.Wrap(sdkerrors.ErrInvalidVersion,
↳ strconv.FormatUint(delegate.Version, 10))
    }

```

```
18      ibcDenom := m.keeper.icaControlKeeper.GetIBCHashDenom(zoneInfo
↳ .TransferInfo.PortId, zoneInfo.TransferInfo.ChannelId, zoneInfo.
↳ BaseDenom)
19
20      versionInfo := m.keeper.GetDelegateVersion(ctx, zoneInfo.
↳ ZoneId)
21      version := versionInfo.Record[delegate.Version]
22      if version.State == types.IcaPending {
23          ok = m.keeper.ChangeDepositState(ctx, zoneInfo.ZoneId,
↳ types.DepositSuccess, types.DelegateRequest)
24          if !ok {
25              return nil, types.ErrCanNotChangeState
26          }
27      }
28
29      delegateAmt := m.keeper.GetTotalDepositAmtForZoneId(ctx,
↳ delegate.ZoneId, ibcDenom, types.DelegateRequest)
30      delegateAmt.Denom = zoneInfo.BaseDenom
31
32      var msgs []sdk.Msg
33      msgs = append(msgs, &stakingtype.MsgDelegate{DelegatorAddress:
↳  zoneInfo.IcaAccount.HostAddress, ValidatorAddress: zoneInfo.
↳ ValidatorAddress, Amount: delegateAmt})
34
35      err := m.keeper.icaControlKeeper.SendTx(ctx, zoneInfo.
↳ IcaConnectionInfo.PortId, zoneInfo.IcaConnectionInfo.ConnectionId,
↳  msgs)
36      if err != nil {
37          return nil, types.ErrDelegateFail
38      }
39
40      versionInfo.Record[delegate.Version] = &types.IBCTrace{
41          Version: versionInfo.CurrentVersion,
42          State:   types.IcaRequest,
43      }
44
45      if err := ctx.EventManager().EmitTypedEvent(
46          types.NewEventDelegate(
47              zoneInfo.IcaAccount.HostAddress,
48              zoneInfo.ValidatorAddress,
49              &delegateAmt,
50              zoneInfo.TransferInfo.ChannelId,
51              zoneInfo.TransferInfo.PortId)); err != nil {
52          return nil, err
```

```
53      }
54
55      return &types.MsgDelegateResponse{}, nil
56  }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to check the amount is greater than zero.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit d5d7903ca by checking for zero amount.

3.15 (HAL-15) LACK OF SIMULATION AND FUZZING OF THE MODULE INVARIANT - LOW

Description:

The **NOVA** system lacks comprehensive CosmosSDK simulations and invariants for its **all** modules. More complete use of the simulation feature would make it easier to fuzz test the entire blockchain and help ensure that invariants hold.

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

Eventually, extend the simulation module to cover all operations that can occur in a real NOVA deployment, along with all possible error states, and run it many times before each release. Make sure of the following:

- All module operations are included in the simulation module.
- The simulation uses some accounts (e.g., between 5 and 20) to increase the likelihood of an interesting state change.
- The simulation uses the currencies/tokens that will be used in the production network.
- The simulation continues to run when a transaction fails.
- All paths of the transaction code are executed. (Enable code coverage to see how often individual lines are executed.)

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of this issue. They will fix it in a future release.

3.16 (HAL-16) LACK OF EVENT EMISSION IS BAD PRACTICE - LOW

Description:

Icacontrol and some of the GAL messages do not currently emit any events. Emitting events is a best practice as it allows off-chain subscribers/indexers to track events.

Code Location:

/x/icacontrol/keeper/msg_server.go#L32-99

Listing 19

```
1 func (k msgServer) RegisterZone(goCtx context.Context, zone *types
↳ .MsgRegisterZone) (*types.MsgRegisterZoneResponse, error)
2 func (k msgServer) ChangeRegisteredZone(goCtx context.Context,
↳ zone *types.MsgChangeRegisteredZone) (*types.
↳ MsgChangeRegisteredZoneResponse, error)
```

/x/gal/keeper/ibc_transfer.go#L21

Listing 20

```
1 func (k msgServer) RegisterZone(goCtx context.Context, zone *types
↳ .MsgRegisterZone) (*types.MsgRegisterZoneResponse, error)
2 func (k msgServer) ChangeRegisteredZone(goCtx context.Context,
↳ zone *types.MsgChangeRegisteredZone) (*types.
↳ MsgChangeRegisteredZoneResponse, error)
```

Recommendation:

Make sure all critical functionality emits events to track operations.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit e962f0b090 by adding events.

3.17 (HAL-17) LACK OF SPEC ON THE MODULES - LOW

Description:

The spec file is intended to outline the common structure for the specifications within this directory. Specifications are missing from **all** NOVA modules. This documentation is segmented into messages focused on the developer and messages directed at the end user. These messages can be displayed to the end user (the human) at the time they will interact with the module.

Risk Level:

Likelihood - 1
Impact - 3

Code Location:

Specs

Recommendation:

It is recommended that modules be fully annotated using specifications for all available functionality.

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of this issue. They will fix it in a future release.

3.18 (HAL-18) COMPUTATIONALLY HEAVY OPERATIONS IN BEGINBLOCKER MAY SLOW DOWN OR STOP BLOCK PRODUCTION - LOW

Description:

BeginBlocker and **EndBlocker** are a way for module developers to add automatic execution of logic to their module. This is a powerful tool that should be used carefully, as complex automation functions can slow down or even halt the chain. There is a module within the scope of this audit where **BeginBlocker** or **EndBlocker** contains unbounded loops that can slow down or even halt the chain.

Code Location:

ABCI Minting Module

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended reworking the **BeginBlocker** and **Endblocker** functions in order to reduce their computational complexity.

Remediation Plan:

SOLVED: The A41 team solved the issue by calculating the complexity of all abci operations.

3.19 (HAL-19) TEST DOCKER IMAGE RUNNING AS ROOT - LOW

Description:

Docker containers generally run with root privileges by default. This allows for unrestricted container management, meaning a user could install system packages, edit configuration files, bind privileged ports, etc. During static analysis, it was observed that the docker image is maintained through the root user.

Code Location:

Dockerfile

Listing 21

```
 1 FROM golang:1.18.1-alpine as build
 2
 3 # this comes from standard alpine nightly file
 4 #  https://github.com/rust-lang/docker-rust-nightly/blob/master/
↳ alpine3.12/Dockerfile
 5 # with some changes to support our toolchain, etc
 6 RUN set -eux; apk add --no-cache ca-certificates build-base;
 7 RUN apk add git
 8
 9 WORKDIR /nova
10 COPY . /nova
11
12 # See https://github.com/CosmWasm/wasmvm/releases
13 ADD https://github.com/CosmWasm/wasmvm/releases/download/v1.0.0/
↳ libwasmvm_muslc.aarch64.a /lib/libwasmvm_muslc.aarch64.a
14 ADD https://github.com/CosmWasm/wasmvm/releases/download/v1.0.0/
↳ libwasmvm_muslc.x86_64.a /lib/libwasmvm_muslc.x86_64.a
15 RUN sha256sum /lib/libwasmvm_muslc.aarch64.a | grep 7
↳ d2239e9f25e96d0d4daba982ce92367aacf0cbd95d2facb8442268f2b1cc1fc
16 RUN sha256sum /lib/libwasmvm_muslc.x86_64.a | grep
↳ f6282df732a13dec836cda1f399dd874b1e3163504dbd9607c6af915b2740479
17 RUN cp /lib/libwasmvm_muslc.$(uname -m).a /lib/libwasmvm_muslc.a
```

```
18 RUN LEDGER_ENABLED=false BUILD_TAGS=muslc LINK_STATICALLY=true
↳ make build
19
20 ## Deploy image
21 FROM golang:1.18.1-alpine
22
23 COPY --from=build /nova/build/novad /bin/novad
24
25 ENV HOME /nova
26 WORKDIR $HOME
27
28 EXPOSE 26656
29 EXPOSE 26657
30 EXPOSE 1317
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to build the Dockerfile and run the container as a non-root user.

Listing 22: Reference

```
1 USER 1001: this is a non-root user UID, and here it is assigned to
↳  the image to run the current container as an unprivileged user.
↳ By doing so, the added security and other restrictions mentioned
↳ above are applied to the container.
```

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of issue. They will fix it in a future release.

3.20 (HAL-20) CSWASM CONTRACT ADDRESS IS NOT VALIDATED ON THE GENESIS STATE - LOW

Description:

The poolincentive module is responsible for the incentives for decentralized exchanges to trade equity tokens and native Supernova coins. This module manages the pool by dividing it into candidate pool and incentive pool. Incentive pools are rewarded with some of the newly issued Nova coin, and users who provide liquidity to the pools can receive them. However, the module does not validate the contract address in the genesis state.

Code Location:

/x/poolincentive/types/genesis.go#L24

Listing 23

```
1
2 func ValidateGenesis(gs GenesisState) error {
3     if err := gs.Params.Validate(); err != nil {
4         return err
5     }
6     return nil
7 }
8
9 func (ip IncentivePool) ValidateBasic() error {
10     // TODO : validate contract address is a valid cosm-wasm
↳ contract address.
11     return nil
12 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to ensure that the contract is checked in the genesis state.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 4cd5df7 by adding validation in genesis.

3.21 (HAL-21) LACK OF VERSION RECORD EXISTENCE CHECK - LOW

Description:

GetDelegateVersion and **GetUndelegateVersion** return the delegation/undelegation version for the zone-id records. However, checking for the existence of records is not implemented in the related functions.

Code Location:

/x/gal/keeper/hooks.go#L96

Listing 24

```
func (h Hooks) AfterDelegateEnd(ctx sdk.Context, delegateMsg
↳ stakingtypes.MsgDelegate) {
    // getZoneInfoForValidatorAddr
    zoneInfo := h.k.icaControlKeeper.
↳ GetRegisteredZoneForValidatorAddr(ctx, delegateMsg.
↳ ValidatorAddress)

    oracleVersion, _ := h.k.oracleKeeper.GetOracleVersion(ctx,
↳ zoneInfo.ZoneId)

    // get delegateVersion
    versionInfo := h.k.GetDelegateVersion(ctx, zoneInfo.ZoneId)
    currentVersion := versionInfo.CurrentVersion

    // change deposit state (DELEGATE_REQUEST -> DELEGATE_SUCCESS)
    h.k.ChangeDepositState(ctx, zoneInfo.ZoneId, types.
↳ DelegateRequest, types.DelegateSuccess)
    h.k.SetDepositOracleVersion(ctx, zoneInfo.ZoneId, types.
↳ DelegateSuccess, oracleVersion)
    h.k.SetDelegateRecordVersion(ctx, zoneInfo.ZoneId, types.
↳ DelegateSuccess, currentVersion)

    versionInfo.Record[currentVersion] = &types.IBCTrace{
        Height: uint64(ctx.BlockHeight()),
        State:  types.IcaSuccess,
```

```
19      }
20
21      nextVersion := versionInfo.CurrentVersion + 1
22      versionInfo.CurrentVersion = nextVersion
23      versionInfo.Record[nextVersion] = &types.IBCTrace{
24          Version: nextVersion,
25          State:   types.IcaPending,
26      }
27      h.k.SetDelegateVersion(ctx, zoneInfo.ZoneId, versionInfo)
28  }
```

/x/gal/keeper/hooks.go#L188

Listing 25

```
 1 func (h Hooks) AfterDelegateFail(ctx sdk.Context, delegateMsg
↳ stakingtypes.MsgDelegate) {
 2     zone := h.k.icaControlKeeper.GetRegisteredZoneForValidatorAddr
↳ (ctx, delegateMsg.ValidatorAddress)
 3
 4     versionInfo := h.k.GetDelegateVersion(ctx, zone.ZoneId)
 5     currentVersion := versionInfo.CurrentVersion
 6
 7     versionInfo.Record[currentVersion] = &types.IBCTrace{
 8         Height:  uint64(ctx.BlockHeight()),
 9         Version: types.IcaFail,
10     }
11
12     h.k.SetDelegateVersion(ctx, zone.ZoneId, versionInfo)
13 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Consider checking for the existence of logs in related functions.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 723721 by adding validation on version.

3.22 (HAL-22) CONFLICT BETWEEN UNDELEGATE AND TRANSFER FAIL HOOKS - LOW

Description:

In the IBC, modules can commit an acknowledgement when receiving and processing a packet in the case of synchronous packet processing. If a packet is processed sometime after the packet is received (asynchronous execution), the acknowledgement will be written after the packet has been processed by the application, which can be long after the packet is received. With messages, the sender's module can process the acknowledgement using the **OnAcknowledgementPacket** callback. In the **gal** module, **Undelegate** and **Transfer** fail hooks use the same storage.

Code Location:

/x/gal/keeper/hooks.go#L219

Listing 26

```
func (h Hooks) AfterUndelegateFail(ctx sdk.Context, undelegateMsg
↳ stakingtypes.MsgUndelegate) {
    zone := h.k.icaControlKeeper.GetRegisteredZoneForValidatorAddr
↳ (ctx, undelegateMsg.ValidatorAddress)

    versionInfo := h.k.GetUndelegateVersion(ctx, zone.ZoneId)
    currentVersion := versionInfo.CurrentVersion

    versionInfo.Record[currentVersion] = &types.IBCTrace{
        Height:  uint64(ctx.BlockHeight()),
        Version: types.IcaFail,
    }

    h.k.SetUndelegateVersion(ctx, zone.ZoneId, versionInfo)
}

```

```
15 func (h Hooks) AfterTransferFail(ctx sdk.Context, transferMsg
↳ transfertypes.MsgTransfer) {
16     zone, ok := h.k.icaControlKeeper.GetRegisterZoneForHostAddr(
↳ ctx, transferMsg.Sender)
17     if !ok {
18         return
19     }
20
21     versionInfo := h.k.GetUndelegateVersion(ctx, zone.ZoneId)
22     currentVersion := versionInfo.CurrentVersion
23
24     versionInfo.Record[currentVersion] = &types.IBCTrace{
25         Height:  uint64(ctx.BlockHeight()),
26         Version: types.IcaFail,
27     }
28
29     h.k.SetUndelegateVersion(ctx, zone.ZoneId, versionInfo)
30 }
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

Consider changing storage and setter function in the **Transfer** and **Undelegate** error hooks.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 145d49a0 by adding a function for **Transfer** and **Undelegate** hooks.

3.23 (HAL-23) MISSING UPPER BOUND DEFINITION ON THE MAX ENTRIES - LOW

Description:

To request the release of the delegation, the equity token must be incinerated. Information about the equity token withdrawn and the actual amount of the withdrawal will be recorded until the withdrawal is completed. During **undelegation**, the maximum entries are checked. However, there is no upper bound on **MaxEntries** during zone registration. The undelegate message contains unbounded storage access on records that reads all stored records, which could cause calling contracts to run out of gas.

Code Location:

/x/icacontrol/types/msgs.go#L72

Listing 27

```
1 func (msg MsgRegisterZone) ValidateBasic() error {
2     if strings.TrimSpace(msg.ZoneId) == "" {
3         return sdkerrors.Wrapf(ErrZoneIdNotNil, "zoneId is not nil
↳ ")
4     }
5
6     if strings.TrimSpace(msg.IcaInfo.ConnectionId) == "" {
7         return errors.New("missing ICA connection ID")
8     }
9
10    _, err := sdk.AccAddressFromBech32(msg.IcaAccount.
↳ ControllerAddress)
11    if err != nil {
12        return sdkerrors.Wrapf(sdkerrors.ErrInvalidAddress, "
↳ Invalid controller address")
13    }
14
```

```
15      if strings.TrimSpace(msg.ValidatorAddress) == "" {
16          return errors.New("missing validator address")
17      }
18
19      if strings.TrimSpace(msg.BaseDenom) == "" {
20          return errors.New("missing denom")
21      }
22
23      if msg.MaxEntries == 0 {
24          return errors.New("cannot set max_entries to zero")
25      }
26
27      return nil
28  }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to define an upper bound on the parameter.

Remediation Plan:

RISK ACCEPTED: The A41 team accepted the risk of this finding.

3.24 (HAL-24) MODULES DO NOT USE REST CLI HANDLER - INFORMATIONAL

Description:

During code review, it was noted that **CosmosSDK** REST handler is not used in modules.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Evaluate if the module needs the CosmosSDK REST interface. This package provides HTTP types and primitives for REST request validation and response handling.

Remediation Plan:

ACKNOWLEDGED: The A41 team acknowledged the issue.

3.25 (HAL-25) OPEN TODOs - INFORMATIONAL

Description:

Open To-dos can point to architectural or programming issues that still need to be resolved. These types of comments often indicate areas of complexity or confusion for developers. This provides value and insight to an attacker who is aiming to cause damage to the protocol.

Code Location:

Listing 28: Open Todos

```
./poolincentive/types/genesis.go:25:    // TODO : validate
↳ contract address is a valid cosm-wasm contract address.
./oracle/module.go:143: // TODO
./oracle/module.go:148: return nil // TODO
./oracle/module.go:153: return nil // TODO
./oracle/module.go:158: // TODO
./icacontrol/spec/en/05_client.md:7:TODO : Now, icacontrol module
↳ not serves query.
./icacontrol/keeper/grpc_query.go:20:   //TODO implement me
./gal/simulation/operations.go:14:  // TODO : implements this!
./gal/simulation/genesis.go:7:  // TODO : implements this!
./gal/keeper/invariants.go:7:   // TODO : implements this!
./gal/module.go:67: // TODO : implements this!
./airdrop/spec/en/05_client.md:5:TODO
./airdrop/keeper/user_state.go:16:  QuestTypeNothingToDo      =
↳ int32(types.QuestType_QUEST_NOTHING_TO_DO)
./airdrop/keeper/msg_server_test.go:22:         questType: types.
↳ QuestType(keeper.QuestTypeNothingToDo),
./airdrop/keeper/msg_server_test.go:33:         questType: types.
↳ QuestType(keeper.QuestTypeNothingToDo),
./airdrop/module.go:159:    // TODO
./airdrop/module.go:164:    return nil // TODO
./airdrop/module.go:169:    return nil // TODO
./airdrop/module.go:174:    // TODO
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Consider resolving the To-dos before deploying code to a production context. Use a standalone issue tracker or other project management software to keep track of development tasks.

Remediation Plan:

ACKNOWLEDGED: The A41 team acknowledged the issue.

3.26 (HAL-26) PANIC IS USED FOR ERROR HANDLING - INFORMATIONAL

Description:

Multiple instances of the panic function have been identified in the codebase. They seem to be used to handle errors. This can cause potential issues, as invoking a panic can cause the program to halt execution and crash in some cases. This, in turn, can negatively affect the availability of the software to users.

Code Location:

Listing 29: Instances of panic identified in the codebase

```
./mint/keeper/keeper.go:34:     panic("the mint module account has not been set")
./mint/keeper/keeper.go:65:     panic(err)
./mint/keeper/keeper.go:81:     panic("stored minter should not have been nil")
./mint/keeper/abci.go:32:       panic(err)
./mint/keeper/abci.go:37:       panic(err)
./mint/module.go:76:        panic(fmt.Sprintf("could not register grpc gateway routes: %v", err))
./poolincentive/keeper/genesis.go:13:           panic(fmt.Errorf("failed to initialize genesis state at %s, err: %v", types.ModuleName, err))
./poolincentive/keeper/genesis.go:19:           panic(fmt.Errorf("failed to initialize genesis state at %s, err: %v", types.ModuleName, err))
./poolincentive/module.go:76:       panic(fmt.Sprintf("could not register grpc gateway routes: %v", err))
./oracle/types/msgs.go:20:      panic(err)
./oracle/keeper/genesis.go:15:          panic(fmt.Errorf("failed to initialize genesis state at %s, err: %v", types.ModuleName, err))
./oracle/keeper/genesis.go:39:          panic(fmt.Errorf("unable to unmarshal chain state: %v", err))
./icacontrol/types/msgs.go:79:      panic(err)
./icacontrol/types/msgs.go:121:     panic(err)
```

```
./icacontrol/types/msgs.go:163:     panic(err)
./icacontrol/types/msgs.go:208:     panic(err)
./icacontrol/types/msgs.go:262:     panic(err)
./icacontrol/types/msgs.go:290:     panic(err)
./icacontrol/types/msgs.go:332:     panic(err)
./icacontrol/types/msgs.go:389:     panic(err)
./icacontrol/types/msgs.go:426:     panic(err)
./icacontrol/keeper/keeper.go:30:       panic(fmt.Sprintf("%s
↳ module account has not been set", types.ModuleName))
./icacontrol/keeper/keeper.go:60:       panic("cannot set ICA
↳ hooks twice")
./icacontrol/keeper/grpc_query.go:21:   panic("implement me")
./icacontrol/client/cli/tx.go:119:              panic("coin error"
↳ )
./gal/keeper/claim.go:28:       panic(fmt.Sprintf("too much
↳ precision, maximum %v, provided %v", snAssetDecimal, prec))
./gal/keeper/claim.go:35:       panic(fmt.Sprintf("too much
↳ precision, maximum %v, provided %v", snAssetDecimal, prec))
./gal/client/cli/tx.go:83:              panic(fmt.Sprintf("can't
↳ parse coin: %s", err.Error()))
./gal/module.go:69:     panic(err)
./airdrop/types/msgs.go:37:     panic(err)
./airdrop/types/msgs.go:61:     panic(err)
./airdrop/types/msgs.go:85:     panic(err)
./airdrop/keeper/action.go:69:      panic("this quest type is not
↳ supported")
./airdrop/keeper/user_state.go:76:                  panic("invalid
↳  claimed amount")
./airdrop/keeper/user_state.go:90:      panic("invalid total
↳ amount")
./airdrop/keeper/airdrop_info_test.go:34:   // check if panic is
↳ thrown when airdrop info is not set
./airdrop/keeper/airdrop_info.go:19:        panic("airdrop info is
↳  missing")
./airdrop/keeper/genesis.go:17:         panic(err)
./airdrop/keeper/genesis.go:21:         panic(err)
./airdrop/module.go:70:     panic(err)
```

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

Instead of using panics, custom errors should be defined and handled according to Best practices.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 71d2cf4 by changing SDK error panics.

3.27 (HAL-27) TYPO ON THE MODULE NAME - INFORMATIONAL

Description:

During code review, it has been noted that the name of the module has a typo.

Code Location:

/x/icacontrol/ibc_mobule.go

Listing 30

```
1 /x/icacontrol/ibc_mobule.go
```

Recommendation:

Consider changing the ibc_mobule.go with ibc_module.go.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit afd74aab by changing the module name.

3.28 (HAL-28) USE OF DEBUG LOGGER INSTEAD OF ERROR - INFORMATIONAL

Description:

During the code review, it was noted that Debugger interface was used instead of error.

Code Location:

Listing 31

```
./x/gal/keeper/grpc_query.go:90:        ctx.Logger().Debug("failed
↳  to find withdraw record", "request", request)
./x/airdrop/keeper/msg_server.go:97:        ctx.Logger().Debug("
↳ user cannot perform the quest anymore")
./x/airdrop/keeper/msg_server.go:102:       ctx.Logger().Debug("
↳ invalid controller address", "addr", signer)
./x/airdrop/keeper/msg_server.go:126:       ctx.Logger().Debug("
↳ user cannot perform the quest anymore")
./x/airdrop/keeper/msg_server.go:131:       ctx.Logger().Debug("
↳ invalid controller address", "addr", signer)
```

Recommendation:

Consider using the log error interface.

Remediation Plan:

ACKNOWLEDGED: The A41 team acknowledged the issue.

3.29 (HAL-29) LACK OF EXTENSIVE TEST COVERAGE - INFORMATIONAL

Description:

Adequate test coverage and regular reporting is an essential process to ensure the codebase works as intended. Insufficient code coverage can lead to unexpected issues and regressions due to changes in module implementations.

Code Location:

Example

Recommendation:

Make sure that the coverage report produced via **go test -cover** covers all functions.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 24524cad by increasing test coverage.

3.30 (HAL-30) UNUSED CODE NEGATIVELY IMPACTS MAINTAINABILITY - INFORMATIONAL

Description:

The codebase contains unused code. Unused code increases the size of the code and thus inhibits maintainability.

Code Location:

Listing 32

```
poolincentive/keeper/keeper.go:38:17: func Keeper.
↳ getCandidatePoolStore is unused
poolincentive/keeper/keeper.go:46:17: func Keeper.
↳ getIncentivePoolInfoStore is unused
poolincentive/keeper/keeper.go:51:17: func Keeper.isValidOperator
↳ is unused
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended removing unused code.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit b2fb050 by deleting unused code.

AUTOMATED TESTING

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped component. Among the tools used were staticcheck, gosec, semgrep, unconvert, LGTM and Nancy. After Halborn verified all the contracts and scoped structures in the repository and was able to compile them correctly, these tools were leveraged on scoped structures. With these tools, Halborn statically verified security related issues across the entire codebase.

Semgrep - Security Analysis Output Sample:

Listing 33: Rule Set

```
semgrep --config "p/dgryski.semgrep-go" x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o dgryski.semgrep
semgrep --config "p/owasp-top-ten"      x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o owasp-top-ten.semgrep
semgrep --config "p/r2c-security-audit" x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o r2c-security-audit.semgrep
semgrep --config "p/r2c-ci"             x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o r2c-ci.semgrep
semgrep --config "p/ci"                 x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o ci.semgrep
semgrep --config "p/golang"             x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o golang.semgrep
semgrep --config "p/trailofbits"        x --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o trailofbits.semgrep
```

Semgrep Results:

Listing 34

```
  docker-compose.yml
     yaml.docker-compose.security.no-new-privileges.no-new-privileges
        Service 'novachain' allows for privilege escalation via setuid or setgid binaries. Add 'no-
```

```
        new-privileges:true' in 'security_opt' to prevent this.
        Details: https://sg.run/0n8q

          3 novachain:
          ----------------------------------------
     yaml.docker-compose.security.no-new-privileges.no-new-
↳ privileges
        Service 'nova-explorer' allows for privilege escalation
↳ via setuid or setgid binaries. Add
        'no-new-privileges:true' in 'security_opt' to prevent this
↳ .
        Details: https://sg.run/0n8q

         19 nova-explorer:
          ----------------------------------------
     yaml.docker-compose.security.writable-filesystem-service.
↳ writable-filesystem-service
        Service 'novachain' is running with a writable root
↳ filesystem. This may allow malicious
        applications to download and run additional payloads, or
↳ modify container files. If an
        application inside a container has to save something
↳ temporarily consider using a tmpfs. Add
        'read_only: true' to this service to prevent this.
        Details: https://sg.run/e4JE

          3 novachain:
          ----------------------------------------
     yaml.docker-compose.security.writable-filesystem-service.
↳ writable-filesystem-service
        Service 'nova-explorer' is running with a writable root
↳ filesystem. This may allow malicious
        applications to download and run additional payloads, or
↳ modify container files. If an
        application inside a container has to save something
↳ temporarily consider using a tmpfs. Add
        'read_only: true' to this service to prevent this.
        Details: https://sg.run/e4JE

         19 nova-explorer:

  x/airdrop/keeper/msg_server.go
     trailofbits.go.invalid-usage-of-modified-variable.invalid-
↳ usage-of-modified-variable
```

```
36          Variable `addr` is likely modified and later used on error
↳ . In some cases this could result
37          in panics due to a nil dereference
38          Details: https://sg.run/WWQ2
39
40          107 addr, err := sdk.AccAddressFromBech32(userAddr)
41          108 if err != nil {
42          109     return nil, sdkerrors.Wrapf(sdkerrors.
↳ ErrInvalidAddress, "addr: %v", addr)
43          110 }
44            ----------------------------------------
45          136 addr, err := sdk.AccAddressFromBech32(userAddr)
46          137 if err != nil {
47          138     return nil, sdkerrors.Wrapf(sdkerrors.
↳ ErrInvalidAddress, "addr: %v", addr)
48          139 }
49
50
51
```

Gosec - Security Analysis Output Sample:

Listing 35

```
Summary:
  Gosec  : v2.14.0
  Files  : 165
  Lines  : 62676
  Nosec  : 0
  Issues : 0
```

Staticcheck - Security Analysis Output Sample:

Listing 36

```
poolincentive/keeper/keeper.go:38:17: func Keeper.
↳ getCandidatePoolStore is unused (U1000)
poolincentive/keeper/keeper.go:46:17: func Keeper.
↳ getIncentivePoolInfoStore is unused (U1000)
poolincentive/keeper/keeper.go:51:17: func Keeper.isValidOperator
↳ is unused (U1000)
```

THANK YOU FOR CHOOSING

// HALBORN