Persistence - Liquid Staking Module

Cosmos Security Assessment

Prepared by: **Halborn**

Date of Engagement: **August 3rd, 2023 - August 22nd, 2023**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY 8

CONTACTS 8

1 EXECUTIVE OVERVIEW 9

1.1 INTRODUCTION 10

1.2 ASSESSMENT SUMMARY 10

1.3 SCOPE 11

2 RISK METHODOLOGY 12

2.1 EXPLOITABILITY 13

2.2 IMPACT 14

2.3 SEVERITY COEFFICIENT 16

2.4 TEST APPROACH & METHODOLOGY 18

3 ASSESSMENT SUMMARY & FINDINGS OVERVIEW 19

4 FINDINGS & TECH DETAILS 21

4.1 (HAL-01) HANDLING OF UNSUCCESSFUL ICA ACCOUNT REGISTRATION IN DORECREATEICA FUNCTION LEADS TO INCONSISTENT CHANNEL STATE - HIGH(8.6) 23

Description 23

Code Location 23

Proof Of Concept 25

BVSS 25

Recommendation 25

Remediation Plan 25

4.2 (HAL-02) COMPUTATIONALLY HEAVY OPERATIONS IN BEGINBLOCKER MAY SLOW DOWN - HIGH(7.8) 26

Description 26

Code Location 26

BVSS 27

Recommendation 27

Remediation Plan 27

4.3 (HAL-03) REPLACEMENT OF PANIC WITH PROPER ERROR HANDLING IN IBCMODULE FUNCTIONS - MEDIUM(6.2) 28

Description 28

Code Location 28

Proof Of Concept 29

BVSS 29

Recommendation 29

Remediation Plan 29

4.4 (HAL-04) IMPLEMENTATION OF IBC RATE LIMIT MODULE FOR PSTAKE CHAIN TO ENHANCE ASSET PROTECTION - MEDIUM(6.2) 31

Description 31

Code Location 31

BVSS 32

Recommendation 32

Remediation Plan 32

4.5 (HAL-05) FAILURE TO DELETE LSCOSMOS MODULE DURING UPGRADE - MEDIUM(4.7) 33

Description 33

Code Location 33

BVSS 34

Recommendation 34

Remediation Plan 34

4.6 (HAL-06) TOKENIZATION OF SHARES ON TOMBSTONED VALIDATORS ALLOWED - MEDIUM(6.2) 35

Description 35

BVSS 35

Recommendation 35

Remediation Plan 35

4.7 (HAL-07) RISK OF UNAUTHORIZED PRIVILEGED ACTIVITY DUE TO HARD-CODED ADMIN ADDRESS - LOW(3.1) 36

Description 36

Code Location 36

BVSS 36

Recommendation 36

Remediation Plan 37

4.8 (HAL-08) VALIDATOR SLASHING LOGIC MISSING ACTION - LOW(3.1) 38

Description 38

Code Location 38

BVSS 38

Recommendation 39

Remediation Plan 39

4.9 (HAL-09) LACK OF MINIMUM DEPOSIT AMOUNT ENFORCEMENT IN LIQUID-STAKELSM FUNCTION - LOW(3.1) 40

Description 40

Code Location 40

BVSS 40

Recommendation 40

Remediation Plan 41

4.10 (HAL-10) LACK OF SPEC ON THE MODULE - LOW(3.1) 42

Description 42

Code Location 42

BVSS 42

Recommendation 42

Remediation Plan 42

4.11 (HAL-11) LACK OF SIMULATION AND FUZZING OF THE MODULE INVARIANT - LOW(3.1) 43

Description 43

Code Location 43

BVSS 43

Recommendation 43

Remediation Plan 44

4.12 (HAL-12) CORRECTION OF DEPOSIT AMOUNT EMISSION IN EVENT ATTRIBUTES FOR ACCURATE TRACKING - LOW(3.1) 45

Description 45

Code Location 45

BVSS 46

Recommendation 46

Remediation Plan 46

4.13 (HAL-13) HARD-CODED CONSTANTS IN LIQUIDSTAKEIBC MODULE - LOW(3.1) 48

Description 48

Code Location 48

BVSS 48

Recommendation 48

Remediation Plan 48

4.14 (HAL-14) OPEN TODOS - INFORMATIONAL(0.0) 49

Description 49

Code Location 49

BVSS 49

Recommendation 50

Remediation Plan 50

4.15 (HAL-15) MISSING USAGE DESCRIPTION FOR ALL TRANSACTION COMMANDS CLI - INFORMATIONAL(0.0) 51

Description 51

Code Location 51

BVSS 52

Recommendation 52

Remediation Plan 52

4.16 (HAL-16) LACK OF VALIDATION FOR UNBONDING FACTOR IN LIQUID-STAKEIBC MODULE - INFORMATIONAL(0.0) 53

Description 53

Code Location 53

BVSS 54

Recommendation 54

Remediation Plan 54

4.17 (HAL-17) LACK OF HANDLING FOR BondStatusUnspecified STATUS IN PROCESSHOSTCHAINVALIDATORUPDATES FUNCTION - INFORMATIONAL(0.0) 55

Description 55

Code Location 55

BVSS 57

Recommendation 57

Remediation Plan 57

4.18 (HAL-18) ABSENCE OF EVENT EMISSION IN LIQUIDSTAKELSM FUNCTION - INFORMATIONAL(0.0) 58

Description 58

Code Location 58

BVSS 58

Recommendation 58

Remediation Plan 59

4.19 (HAL-19) REQUIREMENT OF ADDITIONAL TRANSACTION FOR MODIFYING HARDCODED FLAGS IN REGISTERHOSTCHAIN FUNCTION - INFORMATIONAL(0.0) 60

Description 60

Code Location 60

BVSS 61

Recommendation 61

Remediation Plan 61

5 AUTOMATED TESTING 62

5.1 Description 63

5.2 Semgrep 63

Security Analysis Output Sample 63

Semgrep Results 64

5.3 Gosec 64

Analysis Output Sample 64

5.4 StaticCheck 64

Analysis Output Sample 64

5.5 CodeQL 65

Analysis Output Sample (go and cosmos queries) 65

5.6 Nancy 65

Analysis Output Sample 65

5.7 Codeql 66

Analysis Output Sample 66

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	08/21/2023	Alejandro Taibo
0.2	Document Updates	08/21/2023	Gokberk Gulgun
0.3	Draft Review	08/28/2023	Gabi Urrutia
1.0	Remediation Plan	08/31/2023	Gokberk Gulgun
1.1	Remediation Plan Review	08/31/2023	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
Alejandro Taibo	Halborn	Alejandro.Taibo@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Persistence engaged Halborn to conduct a security assessment on their app chain module beginning on August 3rd, 2023 and ending on August 22nd, 2023. The security assessment was scoped to the liquidstakeibc module provided to the Halborn team.

1.2 ASSESSMENT SUMMARY

The team at Halborn was provided three weeks for the engagement and assigned two full-time security engineers to verify the security of the merge requests. 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 assessment is to:

- Ensure that the **Cosmos Module** operates as intended.
- Identify potential security issues with the staking module.

In summary, Halborn identified some security risks that mostly successfully addressed by the Persistence team.

EXECUTIVE OVERVIEW

1.3 SCOPE

ASSESSMENTS :

IN-SCOPE CODE & COMMIT:

- Repository: pstake-native
    - Commit ID: 633e2f3b8bacf471ebccdd015b5607056c0b7b49
    - Module **in scope**:
        - x/liquidstakeibc.

REMEDIATION COMMIT ID & BRANCH :

- Fix Branch : v2.2.3
  -Commit ID : d90d1d6fc15a4d760d666ce8a4b239c890136231

LSM ASSESSMENT

- Tree: branch
  -Commit ID : ff4cb0314f244e4597d8fe2ec8a5d092a0ee58df

REMEDIATION COMMIT ID & PULL REQUEST :

- Pull Request : 630
  -Commit ID : 370e809878c7538d27b7df2b94a20798e44bd73d

2. RISK METHODOLOGY

Every vulnerability and issue observed by Halborn is ranked based on **two sets** of **Metrics** and a **Severity Coefficient**. This system is inspired by the industry standard Common Vulnerability Scoring System.

The two **Metric sets** are: **Exploitability** and **Impact**. **Exploitability** captures the ease and technical means by which vulnerabilities can be exploited and **Impact** describes the consequences of a successful exploit.

The **Severity Coefficients** is designed to further refine the accuracy of the ranking with two factors: **Reversibility** and **Scope**. These capture the impact of the vulnerability on the environment as well as the number of users and smart contracts affected.

The final score is a value between 0-10 rounded up to 1 decimal place and 10 corresponding to the highest security risk. This provides an objective and accurate rating of the severity of security vulnerabilities in smart contracts.

The system is designed to assist in identifying and prioritizing vulnerabilities based on their level of risk to address the most critical issues in a timely manner.

2.1 EXPLOITABILITY

Attack Origin (AO):

Captures whether the attack requires compromising a specific account.

Attack Cost (AC):

Captures the cost of exploiting the vulnerability incurred by the attacker relative to sending a single transaction on the relevant blockchain. Includes but is not limited to financial and computational cost.

Attack Complexity (AX):

Describes the conditions beyond the attacker's control that must exist in order to exploit the vulnerability. Includes but is not limited to macro situation, available third-party liquidity and regulatory challenges.

Metrics:

Exploitability Metric (m_E)	Metric Value	Numerical Value
Attack Origin (AO)	Arbitrary (AO:A) Specific (AO:S)	1 0.2
Attack Cost (AC)	Low (AC:L) Medium (AC:M) High (AC:H)	1 0.67 0.33
Attack Complexity (AX)	Low (AX:L) Medium (AX:M) High (AX:H)	1 0.67 0.33

Exploitability E is calculated using the following formula:

$$E = \prod m_e$$

2.2 IMPACT

Confidentiality (C):

Measures the impact to the confidentiality of the information resources managed by the contract due to a successfully exploited vulnerability. Confidentiality refers to limiting access to authorized users only.

Integrity (I):

Measures the impact to integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and veracity of data stored and/or processed on-chain. Integrity impact directly affecting Deposit or Yield records is excluded.

Availability (A):

Measures the impact to the availability of the impacted component resulting from a successfully exploited vulnerability. This metric refers to smart contract features and functionality, not state. Availability impact directly affecting Deposit or Yield is excluded.

Deposit (D):

Measures the impact to the deposits made to the contract by either users or owners.

Yield (Y):

Measures the impact to the yield generated by the contract for either users or owners.

Metrics:

Impact Metric (m_I)	Metric Value	Numerical Value
Confidentiality (C)	None (I:N)	0
	Low (I:L)	0.25
	Medium (I:M)	0.5
	High (I:H)	0.75
	Critical (I:C)	1
Integrity (I)	None (I:N)	0
	Low (I:L)	0.25
	Medium (I:M)	0.5
	High (I:H)	0.75
	Critical (I:C)	1
Availability (A)	None (A:N)	0
	Low (A:L)	0.25
	Medium (A:M)	0.5
	High (A:H)	0.75
	Critical	1
Deposit (D)	None (D:N)	0
	Low (D:L)	0.25
	Medium (D:M)	0.5
	High (D:H)	0.75
	Critical (D:C)	1
Yield (Y)	None (Y:N)	0
	Low (Y:L)	0.25
	Medium: (Y:M)	0.5
	High: (Y:H)	0.75
	Critical (Y:H)	1

Impact I is calculated using the following formula:

$$I = max(m_I) + \frac{\sum m_I - max(m_I)}{4}$$

2.3 SEVERITY COEFFICIENT

Reversibility (R):

Describes the share of the exploited vulnerability effects that can be reversed. For upgradeable contracts, assume the contract private key is available.

Scope (S):

Captures whether a vulnerability in one vulnerable contract impacts resources in other contracts.

Coefficient (C)	Coefficient Value	Numerical Value
Reversibility (r)	None (R:N) Partial (R:P) Full (R:F)	1 0.5 0.25
Scope (s)	Changed (S:C) Unchanged (S:U)	1.25 1

Severity Coefficient C is obtained by the following product:

$$C = rs$$

The Vulnerability Severity Score S is obtained by:

$$S = min(10, EIC * 10)$$

The score is rounded up to 1 decimal places.

Severity	Score Value Range
Critical	9 - 10
High	7 - 8.9
Medium	4.5 - 6.9
Low	2 - 4.4
Informational	0 - 1.9

2.4 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 custom modules. 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 assessment :

- Research into architecture and purpose.
- Static Analysis of security for scoped repository, and imported functions. (e.g., staticcheck, gosec, unconvert, codeql, ineffassign and semgrep)
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.
- Dynamic Analysis on files and modules related to the **Liquid Staking** process.

3. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	2	4	7	6

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) HANDLING OF UNSUCCESSFUL ICA ACCOUNT REGISTRATION IN DORECREATEICA FUNCTION LEADS TO INCONSISTENT CHANNEL STATE	High (8.6)	SOLVED - 08/24/2023
(HAL-02) COMPUTATIONALLY HEAVY OPERATIONS IN BEGINBLOCKER MAY SLOW DOWN	High (7.8)	SOLVED - 08/24/2023
(HAL-03) REPLACEMENT OF PANIC WITH PROPER ERROR HANDLING IN IBCMODULE FUNCTIONS	Medium (6.2)	SOLVED - 08/24/2023
(HAL-04) IMPLEMENTATION OF IBC RATE LIMIT MODULE FOR PSTAKE CHAIN TO ENHANCE ASSET PROTECTION	Medium (6.2)	FUTURE RELEASE
(HAL-05) FAILURE TO DELETE LSCOSMOS MODULE DURING UPGRADE	Medium (4.7)	FUTURE RELEASE
(HAL-06) TOKENIZATION OF SHARES ON TOMBSTONED VALIDATORS ALLOWED	Medium (6.2)	SOLVED - 08/31/2023
(HAL-07) RISK OF UNAUTHORIZED PRIVILEGED ACTIVITY DUE TO HARD-CODED ADMIN ADDRESS	Low (3.1)	SOLVED - 08/24/2023
(HAL-08) VALIDATOR SLASHING LOGIC MISSING ACTION	Low (3.1)	SOLVED - 08/24/2023
(HAL-09) LACK OF MINIMUM DEPOSIT AMOUNT ENFORCEMENT IN LIQUIDSTAKELSM FUNCTION	Low (3.1)	SOLVED - 08/31/2023
(HAL-10) LACK OF SPEC ON THE MODULE	Low (3.1)	SOLVED - 08/24/2023
(HAL-11) LACK OF SIMULATION AND FUZZING OF THE MODULE INVARIANT	Low (3.1)	FUTURE RELEASE
(HAL-12) CORRECTION OF DEPOSIT AMOUNT EMISSION IN EVENT ATTRIBUTES FOR ACCURATE TRACKING	Low (3.1)	SOLVED - 08/24/2023
(HAL-13) HARD-CODED CONSTANTS IN LIQUIDSTAKEIBC MODULE	Low (3.1)	FUTURE RELEASE
(HAL-14) OPEN TODOS	Informational (0.0)	SOLVED - 08/24/2023

(HAL-15) MISSING USAGE DESCRIPTION FOR ALL TRANSACTION COMMANDS CLI	Informational (0.0)	SOLVED - 08/24/2023
(HAL-16) LACK OF VALIDATION FOR UNBONDING FACTOR IN LIQUIDSTAKEIBC MODULE	Informational (0.0)	SOLVED - 08/24/2023
(HAL-17) LACK OF HANDLING FOR BondStatusUnspecified STATUS IN PROCESSHOSTCHAINVALIDATORUPDATES FUNCTION	Informational (0.0)	ACKNOWLEDGED
(HAL-18) ABSENCE OF EVENT EMISSION IN LIQUIDSTAKELSM FUNCTION	Informational (0.0)	SOLVED - 08/31/2023
(HAL-19) REQUIREMENT OF ADDITIONAL TRANSACTION FOR MODIFYING HARDCODED FLAGS IN REGISTERHOSTCHAIN FUNCTION	Informational (0.0)	ACKNOWLEDGED

FINDINGS & TECH DETAILS

4.1 (HAL-01) HANDLING OF UNSUCCESSFUL ICA ACCOUNT REGISTRATION IN DORECREATEICA FUNCTION LEADS TO INCONSISTENT CHANNEL STATE - HIGH (8.6)

Description:

In the abci.go for the DoRecreateICA function, there is an attempt to recreate ICA channels for both delegation and rewards accounts. The code checks whether the channel is closed and not being recreated, and if so, it attempts to register the ICA account using the RegisterICAAccount function.

However, there is a potential issue in the current implementation. If the RegisterICAAccount function call fails (i.e., returns an error), the code still proceeds to set the channel state to types.ICAAccount_ICA_CHANNEL_CREATING. This could lead to an inconsistent state where the channel state indicates that it is being created, even though the registration attempt was unsuccessful.

Code Location:

/x/liquidstakeibc/keeper/abci.go#L164

Listing 1

```go
func (k *Keeper) DoRecreateICA(ctx sdk.Context, hc *types.HostChain) {
    // return early if any of the accounts is currently being recreated
    if (hc.DelegationAccount == nil || hc.RewardsAccount == nil) ||
        (hc.DelegationAccount.ChannelState == types.ICAAccount_ICA_CHANNEL_CREATING ||
            hc.RewardsAccount.ChannelState == types.ICAAccount_ICA_CHANNEL_CREATING) {
```

```
            return
    }

    // if the channel is closed, and it is not being recreated, recreate it
    if !k.IsICAChannelActive(ctx, hc, k.GetPortID(hc.DelegationAccount.Owner)) &&
        hc.DelegationAccount.ChannelState != types.ICAAccount_ICA_CHANNEL_CREATING {
        if err := k.RegisterICAAccount(ctx, hc.ConnectionId, hc.DelegationAccount.Owner); err != nil {
            k.Logger(ctx).Error("error recreating %s delegate ica: %w", hc.ChainId, err)
        }

        k.Logger(ctx).Info("Recreating delegate ICA.", "chain", hc.ChainId)

        hc.DelegationAccount.ChannelState = types.ICAAccount_ICA_CHANNEL_CREATING
        k.SetHostChain(ctx, hc)
    }

    // if the channel is closed, and it is not being recreated, recreate it
    if !k.IsICAChannelActive(ctx, hc, k.GetPortID(hc.RewardsAccount.Owner)) &&
        hc.RewardsAccount.ChannelState != types.ICAAccount_ICA_CHANNEL_CREATING {
        if err := k.RegisterICAAccount(ctx, hc.ConnectionId, hc.RewardsAccount.Owner); err != nil {
            k.Logger(ctx).Error("error recreating %s rewards ica: %w", hc.ChainId, err)
        }

        k.Logger(ctx).Info("Recreating rewards ICA.", "chain", hc.ChainId)

        hc.RewardsAccount.ChannelState = types.ICAAccount_ICA_CHANNEL_CREATING
        k.SetHostChain(ctx, hc)
    }
}
```

Proof Of Concept:

Step 1 : Assume that both the delegation and rewards accounts' channels are closed and not in the ICAAccount_ICA_CHANNEL_CREATING state.
Step 2 : Assume that the RegisterICAAccount function is designed to return an error under certain conditions (e.g., invalid connection ID).
Step 3 : The DoRecreateICA function is called with the context and host chain containing the closed delegation and rewards accounts.
Step 4 : The code checks if the delegation account's channel is closed and not being recreated.
Since the conditions are met, it attempts to register the ICA account using k.RegisterICAAccount.
Step 5 : The RegisterICAAccount function returns an error (e.g., due to an invalid connection ID). An error message is logged, but the code continues to execute.
Step 6 : Despite the error, the code sets the delegation account's channel state to types.ICAAccount_ICA_CHANNEL_CREATING.
Step 7 : The host chain is updated with this inconsistent state. The same process is repeated for the rewards account, potentially leading to a similar inconsistent state. The function completes, leaving the delegation and rewards accounts in an inconsistent state where the channel state indicates creation, even though registration was unsuccessful.

BVSS:

AO:A/AC:L/AX:L/C:L/I:M/A:M/D:N/Y:N/R:N/S:C (8.6)

Recommendation:

To avoid this inconsistent state, the code should only update the channel state if the RegisterICAAccount call is successful.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding if statement.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.2 (HAL-02) COMPUTATIONALLY HEAVY OPERATIONS IN BEGINBLOCKER MAY SLOW DOWN - HIGH (7.8)

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 automatic functions can slow down. There is a one module within the scope of this assessment where the **BeginBlocker** or **EndBlocker** contains unbounded loops that can slow.

The BeginBlock function in the provided code snippet iterates over all host chains and performs several tasks, including recreating ICA channels, delegating, claiming, and processing matured undelegations. While this functionality is essential, the unbounded loop that iterates over all host chains may lead to performance issues, especially if there are many host chains. Complex automatic functions within the BeginBlocker can slow down, impacting the overall performance and stability of the system.

Code Location:

Listing 2

```
func (k *Keeper) BeginBlock(ctx sdk.Context) {

    // perform BeginBlocker tasks for each chain
    for _, hc := range k.GetAllHostChains(ctx) {
        if !hc.Active {
            // don't do anything on inactive chains
            continue
        }

        // attempt to recreate closed ICA channels
        k.DoRecreateICA(ctx, hc)

        // attempt to delegate
        k.DoDelegate(ctx, hc)

        // attempt to automatically claim matured undelegations
        k.DoClaim(ctx, hc)

        // attempt to process any matured unbondings
        k.DoProcessMaturedUndelegations(ctx, hc)
    }
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:M/D:N/Y:N/R:N/S:C (7.8)

Recommendation:

Introduce a mechanism to limit the number of host chains processed in a single block. This can be achieved by setting a configurable threshold or implementing a priority system based on certain criteria.

Remediation Plan:

SOLVED: The Persistence team solved the issue by limiting host chains.

4.3 (HAL-03) REPLACEMENT OF PANIC WITH PROPER ERROR HANDLING IN IBCMODULE FUNCTIONS - MEDIUM (6.2)

Description:

In the implementation of the IBCModule interface, several functions are currently using the panic statement to indicate that they are unimplemented. While this approach may be suitable for early development stages, it can lead to abrupt termination of the program and is generally considered an antipattern in production code.

The following functions within the IBCModule interface are using panic("UNIMPLEMENTED"):

- OnChanOpenTry
- OnChanOpenConfirm
- OnChanCloseInit
- OnChanCloseConfirm
- OnRecvPacket

Code Location:

/x/liquidstakeibc/module_ibc.go#L104

Listing 3

```
func (m IBCModule) OnChanOpenConfirm(ctx sdk.Context, portID, channelID string) error {
    panic("UNIMPLEMENTED")
}

func (m IBCModule) OnChanCloseInit(ctx sdk.Context, portID, channelID string) error {
    panic("UNIMPLEMENTED")
}

func (m IBCModule) OnChanCloseConfirm(ctx sdk.Context, portID,
```

```go
channelID string) error {
    panic("UNIMPLEMENTED")
}

func (m IBCModule) OnRecvPacket(ctx sdk.Context, packet channeltypes.Packet, relayer sdk.AccAddress) ibcexported.Acknowledgement {
    panic("UNIMPLEMENTED")
}
```

Proof Of Concept:

Step 1 : Assuming the OnRecvPacket function is implemented as follows:

Listing 4

```go
func (m IBCModule) OnRecvPacket(ctx sdk.Context, packet channeltypes.Packet, relayer sdk.AccAddress) ibcexported.Acknowledgement {
    panic("UNIMPLEMENTED")
}
```

Step 2 :Now, if any part of the code calls this function, it will result in a panic, abruptly terminating the program.

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:M/D:N/Y:N/R:N/S:C (6.2)

Recommendation:

It is recommended to replace the panic statements with proper error handling.

Remediation Plan:

SOLVED: The Persistence team solved the issue by changing panic with nil.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.4 (HAL-04) IMPLEMENTATION OF IBC RATE LIMIT MODULE FOR PSTAKE CHAIN TO ENHANCE ASSET PROTECTION - MEDIUM (6.2)

Description:

The IBC Rate Limit module, designed to add a governance-configurable rate limit to IBC transfers, serves as a critical safety control to protect assets on chains. It aims to mitigate risks associated with potential bugs or hacks on chains, the counter-party chain, or within IBC itself. While this module offers robust protection, especially for high-value IBC connections, it may also introduce a one-way bridge liveness tradeoff during periods of high deposits or withdrawals.

The Liquid Staking module, which may have similar requirements for asset protection and controlled rate limiting, currently lacks this functionality. Implementing the IBC Rate Limit module could enhance the security and control of asset transfers within the chain.

Code Location:

/x/liquidstakeibc/module_ibc.go

Listing 5

```go
package liquidstakeibc

import (
	sdk "github.com/cosmos/cosmos-sdk/types"
	capabilitytypes "github.com/cosmos/cosmos-sdk/x/capability/types"
	channeltypes "github.com/cosmos/ibc-go/v7/modules/core/04-channel/types"
	porttypes "github.com/cosmos/ibc-go/v7/modules/core/05-port/types"
	ibcexported "github.com/cosmos/ibc-go/v7/modules/core/exported
"

    "github.com/persistenceOne/pstake-native/v2/x/liquidstakeibc/keeper"
)

var _ porttypes.IBCModule = &IBCModule{}

// IBCModule implements the ICS26 callbacks for the fee middleware given the
// fee keeper and the underlying application.
type IBCModule struct {
    keeper keeper.Keeper
}

func NewIBCModule(keeper keeper.Keeper) IBCModule {
    return IBCModule{
        keeper: keeper,
    }
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:M/D:N/Y:N/R:N/S:C (6.2)

Recommendation:

To enhance asset protection on the pstake chain, it is advisable to consider implementing the IBC Rate Limit module. This module can provide governance-configurable rate limits for IBC transfers, mitigating risks associated with potential bugs or hacks.

Remediation Plan:

PENDING: The Persistence team plans to address this issue in an upcoming release.

4.5 (HAL-05) FAILURE TO DELETE LSCOSMOS MODULE DURING UPGRADE - MEDIUM (4.7)

Description:

In the planned system upgrade, the lscosmos module is intended to be replaced with the liquidstake module. However, the code snippet provided shows that the deletion of the lscosmos module is commented out. This means that the lscosmos module will not be deleted during the upgrade, leading to a discrepancy between the intended upgrade plan and the actual implementation.

The presence of both the liquidstake and lscosmos modules can lead to conflicts on the app chain.

Code Location:

/app/app.go#L1080

Listing 6

```
    if upgradeInfo.Name == upgradeName && !app.UpgradeKeeper.IsSkipHeight(upgradeInfo.Height) {
        storeUpgrades := store.StoreUpgrades{
            Added: []string{
                liquidstakeibctypes.ModuleName,
                lspersistencetypes.ModuleName,
                consensusparamtypes.ModuleName,
                crisistypes.ModuleName,
                ibcfeetypes.ModuleName,
            },
            Deleted: []string{
                //lscosmostypes.ModuleName, add this to next upgrade.
            },
        }

        // configure store loader that checks if version == upgradeHeight and applies store upgrades
        app.SetStoreLoader(upgradetypes.UpgradeStoreLoader(upgradeInfo.Height, &storeUpgrades))
    }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:H/A:N/D:N/Y:N/R:P/S:C (4.7)

Recommendation:

It is recommended to uncomment the line of code that deletes the lscosmos module to ensure that it is removed during the upgrade, as intended.

Remediation Plan:

PENDING: The Persistence team plans to address this issue in an upcoming release.

4.6 (HAL-06) TOKENIZATION OF SHARES ON TOMBSTONED VALIDATORS ALLOWED - MEDIUM (6.2)

Description:

The current LSM module allows for the tokenization of shares on a tombstoned validator. This could potentially lead to a situation where users are holding essentially worthless tokenized shares.

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:M/D:N/Y:N/R:N/S:C (6.2)

Recommendation:

Consider implementing a check to prevent the tokenization of shares on tombstoned validators.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding a validator status check.

Commit ID: bd53259036d5b2696076433530b3f4098634e23c

4.7 (HAL-07) RISK OF UNAUTHORIZED PRIVILEGED ACTIVITY DUE TO HARD-CODED ADMIN ADDRESS - LOW (3.1)

Description:

The code snippet provided shows the use of hard-coded admin address within the application, specifically for the DefaultAdminAddress. The hard-coded value increases the risk of unauthorized privileged activity, as they may be more susceptible to compromise. All host chain parameters are managed by the governance module and admin address.

Code Location:

/types/params.go#L8

Listing 7

```go
const (
    DefaultAdminAddress string = "persistence10khgeppewe4rgfrcy809r9h00aquwxxxrk6glr" // TODO: Use correct address on launch
    DefaultFeeAddress   string = "persistence1xruvjju28j0a5ud5325rfdak8f5a04h0s30mld" // TODO: Use correct address on launch
)
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

Consider managing the host chain parameters through a governance module.

Remediation Plan:

SOLVED: The Persistence team solved the issue with changing constant variable with governance parameter.

Commit ID : ff4cb0314f244e4597d8fe2ec8a5d092a0ee58df

4.8 (HAL-08) VALIDATOR SLASHING LOGIC MISSING ACTION - LOW (3.1)

Description:

In the code handling different cases related to validators, there is a specific case for types.KeyValidatorSlashing that seems to lack any concrete action for slashing a validator. While the code checks for the existence of the validator, it does not appear to perform any actual slashing or related operations.

The code retrieves the validator and performs a query, but there is no subsequent action to slash the validator or modify its state in any way.

Code Location:

/x/liquidstakeibc/keeper/msg_server.go#L167

Listing 8

```
    case types.KeyValidatorSlashing:
      _, found = hc.GetValidator(update.Value)
      if !found {
        return nil, types.ErrValidatorNotFound
      }

      if err := k.QueryHostChainValidator(ctx, hc, update.Value); err != nil {
        return nil, fmt.Errorf("unable to send ICQ query for validator")
      }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

To address this issue, it is recommended to implement the necessary logic to perform the slashing operation on the identified validator. This may include:

- Defining Slashing Parameters: Determine the appropriate slashing parameters, such as the percentage of stake to be slashed, based on your protocol's rules and requirements.
- Implementing Slashing Logic: Add the necessary code to modify the validator's state to reflect the slashing. This may involve reducing the validator's stake, updating related metrics, and emitting relevant events.
- Handling Associated Actions: Consider any additional actions that must be taken in conjunction with slashing, such as notifying the validator, updating related accounts, or triggering other protocol mechanisms.

Remediation Plan:

SOLVED: The Persistence team solved the issue by handling slashed validator in the icq.

Commit ID: 52ffa80d298c5d9b7255ba3789c4d4d2db22b1d0

4.9 (HAL-09) LACK OF MINIMUM DEPOSIT AMOUNT ENFORCEMENT IN LIQUIDSTAKELSM FUNCTION - LOW (3.1)

Description:

The LiquidStakeLSM function currently lacks a mechanism to enforce a minimum deposit amount, unlike its counterpart LiquidStake, which has a check for hostChain.MinimumDeposit.

Code Location:

/x/liquidstakeibc/keeper/msg_server.go#L421

Listing 9

```go
func (k msgServer) LiquidStakeLSM(
    goCtx context.Context,
    msg *types.MsgLiquidStakeLSM,
) (*types.MsgLiquidStakeLSMResponse, error) {
    ctx := sdktypes.UnwrapSDKContext(goCtx)}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

To align the LiquidStakeLSM function with the LiquidStake function and to ensure operational integrity, it is recommended to implement a minimum deposit amount check in LiquidStakeLSM.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding a minimum deposit requirement.

Commit ID: 9adb53504e59c5d3cc77ac0f8e3dac20555f919d

4.10 (HAL-10) LACK OF SPEC ON THE MODULE - LOW (3.1)

Description:

The spec file intends to outline the common structure for specifications within this directory. In the pStake's liquidstakeibc module is missing spec. This documentation is segmented into developer-focused messages and end-user-facing messages. These messages may be shown to the end user (the human) at the time that they will interact with the module.

Code Location:

Tree

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

It is recommended that modules are fully annotated using spec for all available functionalities.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding the specs.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.11 (HAL-11) LACK OF SIMULATION AND FUZZING OF THE MODULE INVARIANT - LOW (3.1)

Description:

The Persistence system lacks comprehensive CosmosSDK simulations and invariants for its x/liquidstakeibc module. More thorough use of the simulation feature would facilitate fuzz testing of the entire blockchain and help ensure that the invariants hold.

Code Location:

tree

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

Long term, extend the simulation module to cover all operations that may occur in a real module deployment, along with all potential error states, and run it many times before each release. Ensure the following:

- All module operations are included in the simulation module.
- The simulation uses a few 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 running when a transaction triggers an error.
- All transaction code paths are executed. (Enable code coverage to see how often individual lines are executed.)

Remediation Plan:

PENDING: The Persistence team plans to address this issue in an upcoming release.

4.12 (HAL-12) CORRECTION OF DEPOSIT AMOUNT EMISSION IN EVENT ATTRIBUTES FOR ACCURATE TRACKING - LOW (3.1)

Description:

In the current implementation of the event emission within the deposit process, the attribute AttributeAmount and AttributeAmountReceived are being set with the value of ‘mintToken.Sub(protocolFee).String()‘‘. This value represents the minted tokens minus the protocol fee, which may not accurately reflect the actual deposit amount made by the delegator.

This discrepancy could lead to confusion or incorrect tracking of the deposit amounts in the system, potentially affecting subsequent calculations or reporting related to the deposits.

Code Location:

/x/liquidstakeibc/keeper/msg_server.go#L379

Listing 10

```go
    ctx.EventManager().EmitEvents(sdktypes.Events{
        sdktypes.NewEvent(
            types.EventTypeLiquidStake,
            sdktypes.NewAttribute(types.AttributeDelegatorAddress, delegatorAddress.String()),
            sdktypes.NewAttribute(types.AttributeAmount, mintToken.String()),
            sdktypes.NewAttribute(types.AttributeAmountReceived, mintToken.Sub(protocolFee).String()),
            sdktypes.NewAttribute(types.AttributePstakeDepositFee, protocolFee.String()),
        ),
        sdktypes.NewEvent(
            sdktypes.EventTypeMessage,
            sdktypes.NewAttribute(sdktypes.AttributeKeyModule, types.AttributeValueCategory),
            sdktypes.NewAttribute(sdktypes.AttributeKeySender, msg.DelegatorAddress),
        )},
    )
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

Consider using depositAmount instead of minted token amount.

Listing 11

```
ctx.EventManager().EmitEvents(sdktypes.Events{
    sdktypes.NewEvent(
        types.EventTypeLiquidStake,
        sdktypes.NewAttribute(types.AttributeDelegatorAddress, delegatorAddress.String()),
        sdktypes.NewAttribute(types.AttributeAmount, depositAmount.String()), // Emit depositAmount here
        sdktypes.NewAttribute(types.AttributeAmountReceived, depositAmount.String()), // Emit depositAmount here if needed
        sdktypes.NewAttribute(types.AttributePstakeDepositFee, protocolFee.String()),
    ),
    sdktypes.NewEvent(
        sdktypes.EventTypeMessage,
        sdktypes.NewAttribute(sdktypes.AttributeKeyModule, types.AttributeValueCategory),
        sdktypes.NewAttribute(sdktypes.AttributeKeySender, msg.DelegatorAddress),
    )},
)
```

Remediation Plan:

SOLVED: The Persistence team solved the issue by fixing the event.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.13 (HAL-13) HARD-CODED CONSTANTS IN LIQUIDSTAKEIBC MODULE - LOW (3.1)

Description:

The liquidstakeibc module within the app chain contains several hard-coded values, including constants related to IBC. These hard-coded values can lead to inflexibility, as they cannot be easily updated or managed dynamically. The presence of such constants may limit the adaptability of the module

Code Location:

/types/keys.go#L43

Listing 12

```

    IBCTimeoutHeightIncrement uint64 = 1000

    ICATimeoutTimestamp = 15 * time.Minute
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:P/S:C (3.1)

Recommendation:

To address this issue, consider moving critical constants and parameters to governance-controlled values.

Remediation Plan:

PENDING: The Persistence team plans to address this issue in an upcoming release.

4.14 (HAL-14) OPEN TODOS - INFORMATIONAL (0.0)

Description:

Open To-dos can point to architecture or programming issues that still need to be resolved. Often these kinds of comments indicate areas of complexity or confusion for developers. This provides value and insight to an attacker who aims to cause damage to the protocol.

Code Location:

/app/app.go#L507

Listing 13

```
app.LiquidStakeIBCKeeper = liquidstakeibckeeper.NewKeeper(
    appCodec,
    keys[liquidstakeibctypes.StoreKey],
    app.AccountKeeper,
    app.BankKeeper,
    app.EpochsKeeper,
    app.ICAControllerKeeper,
    app.IBCKeeper, // TODO: Move to module interface
    &app.InterchainQueryKeeper,
    app.GetSubspace(liquidstakeibctypes.ModuleName),
    app.MsgServiceRouter(),
    authtypes.NewModuleAddress(govtypes.ModuleName).String(),
)
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

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

Remediation Plan:

SOLVED: The Persistence team solved the issue by resolving the **TO-DOs**.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.15 (HAL-15) MISSING USAGE DESCRIPTION FOR ALL TRANSACTION COMMANDS CLI - INFORMATIONAL (0.0)

Description:

All the transaction and query commands for the liquidstakeibc module are missing a long message to provide description about their usage, which will be helpful for users and external developers.

Code Location:

/client/tx.go#L22

Listing 14

```
func NewTxCmd() *cobra.Command {
	txCmd := &cobra.Command{
		Use:                        types.ModuleName,
		Short:                      "Pstake liquid staking ibc transaction subcommands",
		DisableFlagParsing:         true,
		SuggestionsMinimumDistance: 2,
		RunE:                       client.ValidateCmd,
	}

	txCmd.AddCommand(
		NewRegisterHostChainCmd(),
		NewUpdateHostChainCmd(),
		NewLiquidStakeCmd(),
		NewLiquidUnstakeCmd(),
		NewRedeemCmd(),
		NewUpdateParamsCmd(),
	)

	return txCmd
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

It is recommended to specify a long message for all transaction commands. Each command should provide a description of how to use the command correctly.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding description.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.16 (HAL-16) LACK OF VALIDATION FOR UNBONDING FACTOR IN LIQUIDSTAKEIBC MODULE - INFORMATIONAL (0.0)

Description:

In the liquidstakeibc module, the NewMsgRegisterHostChain function accepts an unbondingFactor parameter, which is used to create a new MsgRegisterHostChain object. However, within the ValidateBasic method of the MsgRegisterHostChain type, there is no validation performed on the unbondingFactor. This omission can lead to potential issues, as invalid or malicious values for the unbondingFactor could be accepted without any checks.

Code Location:

/x/liquidstakeibc/types/msgs.go#L61C23-L61C38

Listing 15

```
func NewTxCmd() *cobra.Command {
    txCmd := &cobra.Command{
        Use:                        types.ModuleName,
        Short:                      "Pstake liquid staking ibc transaction subcommands",
        DisableFlagParsing:         true,
        SuggestionsMinimumDistance: 2,
        RunE:                       client.ValidateCmd,
    }

    txCmd.AddCommand(
        NewRegisterHostChainCmd(),
        NewUpdateHostChainCmd(),
        NewLiquidStakeCmd(),
        NewLiquidUnstakeCmd(),
        NewRedeemCmd(),
        NewUpdateParamsCmd(),
    )

    return txCmd
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

To address this issue, it is recommended to add appropriate validation for the unbondingFactor within the ValidateBasic method.

Remediation Plan:

SOLVED: The Persistence team solved the issue by adding validation on the unbondingFactor.

Commit ID: d90d1d6fc15a4d760d666ce8a4b239c890136231

4.17 (HAL-17) LACK OF HANDLING FOR BondStatusUnspecified STATUS IN PROCESSHOSTCHAINVALIDATORUPDATES FUNCTION - INFORMATIONAL (0.0)

Description:

The ProcessHostChainValidatorUpdates function in the provided code snippet processes updates to a host chain validator, including changes to the validator's status and exchange rate. The code handles transitions into BondStatusUnbonding, BondStatusUnbonded, and BondStatusBonded statuses. However, there is no mention or handling of a status that might be considered "unspecified." The codebase lacks a definition for the BondStatusUnspecified status within the staking types, and this omission could lead to unexpected behavior or errors if a validator's status were to be in an unspecified state.

Code Location:

host_chain.go#L43C1-L93C2

Listing 16

```go
func (k *Keeper) ProcessHostChainValidatorUpdates(
    ctx sdk.Context,
    hc *types.HostChain,
    validator stakingtypes.Validator,
) error {
    val, found := hc.GetValidator(validator.OperatorAddress)
    if !found {
        return fmt.Errorf("validator with address %s not registered", validator.OperatorAddress)
    }

    // process status update
    if validator.Status.String() != val.Status {
        // validator transitioned into unbonding
        if validator.Status.String() == stakingtypes.BondStatusUnbonding ||
            validator.Status.String() == stakingtypes.BondStatusUnbonded {
            epochNumber := k.epochsKeeper.GetEpochInfo(ctx, types.UndelegationEpoch).CurrentEpoch
            val.UnbondingEpoch = types.CurrentUnbondingEpoch(hc.UnbondingFactor, epochNumber)
        }
        // validator transitioned into bonded
        if validator.Status.String() == stakingtypes.BondStatusBonded {
            val.UnbondingEpoch = 0
        }

        val.Status = validator.Status.String()
        k.SetHostChainValidator(ctx, hc, val)
    }

    // process exchange rate update
    var exchangeRate sdk.Dec
    if validator.DelegatorShares.IsZero() {
        exchangeRate = sdk.OneDec()
    } else {
        exchangeRate = sdk.NewDecFromInt(validator.Tokens).Quo(validator.DelegatorShares)
    }
    if !exchangeRate.Equal(val.ExchangeRate) {
        if val.DelegatedAmount.GT(sdk.ZeroInt()) {
            if err := k.QueryValidatorDelegation(ctx, hc, val); err != nil {
                return fmt.Errorf(
                    "error while querying validator %s delegation: %s",
                    val.OperatorAddress,
                    err.Error(),
                )
            }
        }

        val.ExchangeRate = exchangeRate
        k.SetHostChainValidator(ctx, hc, val)
    }

    return nil
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

Modify the ProcessHostChainValidatorUpdates function to handle the BondStatusUnspecified status appropriately. This might include adding specific logic for this status or handling it as an error condition if it represents an invalid or unexpected state.

Remediation Plan:

ACKNOWLEDGED: The Persistence team acknowledged the issue.

4.18 (HAL-18) ABSENCE OF EVENT EMISSION IN LIQUIDSTAKELSM FUNCTION - INFORMATIONAL (0.0)

Description:

The LiquidStakeLSM function currently lacks event emission, unlike its counterpart LiquidStake which emits events for important actions such as staking, protocol fees, and more. This absence of event logging can make it difficult to track and audit the actions performed by the LiquidStakeLSM function.

Code Location:

/x/liquidstakeibc/keeper/msg_server.go#L408

Listing 17

```
func (k msgServer) LiquidStakeLSM(
    goCtx context.Context,
    msg *types.MsgLiquidStakeLSM,
) (*types.MsgLiquidStakeLSMResponse, error) {}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

It is advisable to include event emission in the LiquidStakeLSM function similar to how it is done in the LiquidStake function. This will improve transparency and make it easier to monitor and debug the system.

Remediation Plan:

SOLVED: The Persistence team solved the issue by omitting the event.

Commit ID: 811808e301822d36ced5e857ee54de415a373fa9

4.19 (HAL-19) REQUIREMENT OF ADDITIONAL TRANSACTION FOR MODIFYING HARDCODED FLAGS IN REGISTERHOSTCHAIN FUNCTION - INFORMATIONAL (0.0)

Description:

The RegisterHostChain function in the LSM module has hardcoded flags, specifically the Lsm: false flag within the HostChainFlags struct. While this does not restrict the ability to enable or disable LSM features, it necessitates an additional transaction to modify these flags, thereby incurring extra gas costs.

Code Location:

/x/liquidstakeibc/keeper/msg_server.go#L81

Listing 18

```
hc := &types.HostChain{
    ChainId:         chainID,
    ConnectionId:    msg.ConnectionId,
    ChannelId:       msg.ChannelId,
    PortId:          msg.PortId,
    Params:          hostChainParams,
    HostDenom:       msg.HostDenom,
    MinimumDeposit:  msg.MinimumDeposit,
    CValue:          sdktypes.NewDec(1),
    UnbondingFactor: msg.UnbondingFactor,
    Active:          false,
    DelegationAccount: &types.ICAAccount{
        Owner:   types.DefaultDelegateAccountPortOwner(chainID),
        Balance: sdktypes.Coin{Amount: sdktypes.ZeroInt(), Denom: msg.HostDenom},
    },
    RewardsAccount: &types.ICAAccount{
        Owner:   types.DefaultRewardsAccountPortOwner(chainID),
        Balance: sdktypes.Coin{Amount: sdktypes.ZeroInt(), Denom: msg.HostDenom},
    },
    AutoCompoundFactor: k.CalculateAutocompoundLimit(sdktypes.NewDec(msg.AutoCompoundFactor)),
    Flags: &types.HostChainFlags{
        Lsm: false,
    },
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:N/R:F/S:C (0.0)

Recommendation:

To improve efficiency and reduce operational complexity, consider refactoring the RegisterHostChain function to allow for dynamic flag settings.

Remediation Plan:

ACKNOWLEDGED: The Persistence team acknowledged the issue.

AUTOMATED TESTING

5.1 Description

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped component. Among the tools used were staticcheck, gosec, semgrep, codeQL 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 can statically verify security related issues across the entire codebase.

5.2 Semgrep

Security Analysis Output Sample:

Listing 19: Rule Set

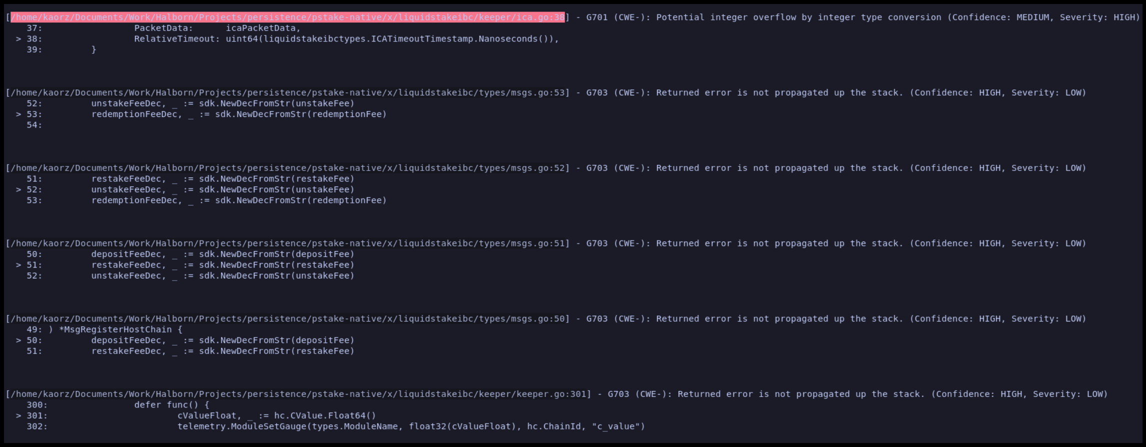
```
semgrep --config "p/dgryski.semgrep-go" x/liquidstakeibc --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o dgryski.semgrep
semgrep --config "p/owasp-top-ten"      x/liquidstakeibc --exclude='*_test.go' --max-lines-per-finding 1000 --no-git-ignore -o owasp-top-ten.semgrep
semgrep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

Semgrep Results:

- No major issues found by Semgrep.

5.3 Gosec

Analysis Output Sample:



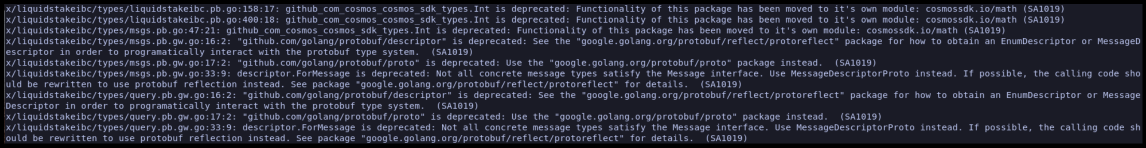
```
[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/keeper/ica.go:38] - G701 (CWE-): Potential integer overflow by integer type conversion (Confidence: MEDIUM, Severity: HIGH)
    37:                 PacketData:      icaPacketData,
  > 38:                 RelativeTimeout: uint64(liquidstakeibctypes.ICATimeoutTimestamp.Nanoseconds()),
    39:         }



[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/types/msgs.go:53] - G703 (CWE-): Returned error is not propagated up the stack. (Confidence: HIGH, Severity: LOW)
    52:         unstakeFeeDec, _ := sdk.NewDecFromStr(unstakeFee)
  > 53:         redemptionFeeDec, _ := sdk.NewDecFromStr(redemptionFee)
    54:



[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/types/msgs.go:52] - G703 (CWE-): Returned error is not propagated up the stack. (Confidence: HIGH, Severity: LOW)
    51:         restakeFeeDec, _ := sdk.NewDecFromStr(restakeFee)
  > 52:         unstakeFeeDec, _ := sdk.NewDecFromStr(unstakeFee)
    53:         redemptionFeeDec, _ := sdk.NewDecFromStr(redemptionFee)



[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/types/msgs.go:51] - G703 (CWE-): Returned error is not propagated up the stack. (Confidence: HIGH, Severity: LOW)
    50:         depositFeeDec, _ := sdk.NewDecFromStr(depositFee)
  > 51:         restakeFeeDec, _ := sdk.NewDecFromStr(restakeFee)
    52:         unstakeFeeDec, _ := sdk.NewDecFromStr(unstakeFee)



[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/types/msgs.go:50] - G703 (CWE-): Returned error is not propagated up the stack. (Confidence: HIGH, Severity: LOW)
    49: ) *MsgRegisterHostChain {
  > 50:         depositFeeDec, _ := sdk.NewDecFromStr(depositFee)
    51:         restakeFeeDec, _ := sdk.NewDecFromStr(restakeFee)



[/home/kaorz/Documents/Work/Halborn/Projects/persistence/pstake-native/x/liquidstakeibc/keeper/keeper.go:301] - G703 (CWE-): Returned error is not propagated up the stack. (Confidence: HIGH, Severity: LOW)
    300:                defer func() {
  > 301:                        cValueFloat, _ := hc.CValue.Float64()
    302:                        telemetry.ModuleSetGauge(types.ModuleName, float32(cValueFloat), hc.ChainId, "c_value")
```

Figure 1: Gosec results

- No major issues found by Gosec.

5.4 StaticCheck

Analysis Output Sample:



```
x/liquidstakeibc/types/liquidstakeibc.pb.go:158:17: github_com_cosmos_cosmos_sdk_types.Int is deprecated: Functionality of this package has been moved to it's own module: cosmossdk.io/math (SA1019)
x/liquidstakeibc/types/liquidstakeibc.pb.go:400:18: github_com_cosmos_cosmos_sdk_types.Int is deprecated: Functionality of this package has been moved to it's own module: cosmossdk.io/math (SA1019)
x/liquidstakeibc/types/msgs.pb.go:47:21: github_com_cosmos_cosmos_sdk_types.Int is deprecated: Functionality of this package has been moved to it's own module: cosmossdk.io/math (SA1019)
x/liquidstakeibc/types/msgs.pb.gw.go:16:2: "github.com/golang/protobuf/descriptor" is deprecated: See the "google.golang.org/protobuf/reflect/protoreflect" package for how to obtain an EnumDescriptor or MessageD
escriptor in order to programatically interact with the protobuf type system.  (SA1019)
x/liquidstakeibc/types/msgs.pb.gw.go:17:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/liquidstakeibc/types/msgs.pb.gw.go:33:9: descriptor.ForMessage is deprecated: Not all concrete message types satisfy the Message interface. Use MessageDescriptorProto instead. If possible, the calling code sho
uld be rewritten to use protobuf reflection instead. See package "google.golang.org/protobuf/reflect/protoreflect" for details.  (SA1019)
x/liquidstakeibc/types/query.pb.gw.go:16:2: "github.com/golang/protobuf/descriptor" is deprecated: See the "google.golang.org/protobuf/reflect/protoreflect" package for how to obtain an EnumDescriptor or Message
Descriptor in order to programatically interact with the protobuf type system.  (SA1019)
x/liquidstakeibc/types/query.pb.gw.go:17:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/liquidstakeibc/types/query.pb.gw.go:33:9: descriptor.ForMessage is deprecated: Not all concrete message types satisfy the Message interface. Use MessageDescriptorProto instead. If possible, the calling code sh
ould be rewritten to use protobuf reflection instead. See package "google.golang.org/protobuf/reflect/protoreflect" for details.  (SA1019)
```

Figure 2: StaticCheck results

- No major issues found by StaticCheck.

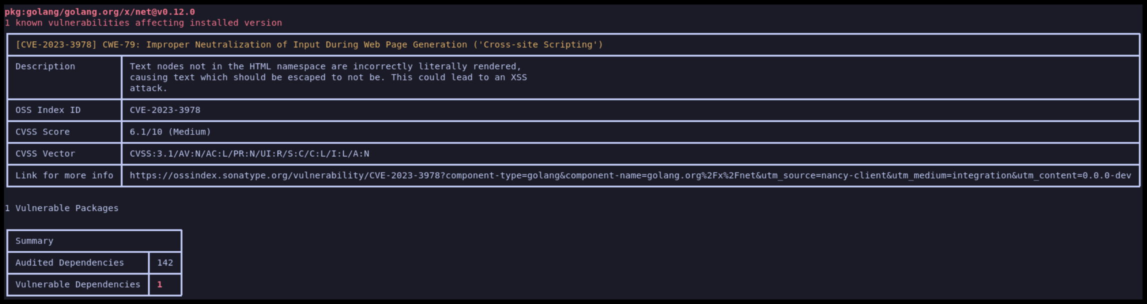
5.5 CodeQL

Analysis Output Sample (go and cosmos queries):

- No major issues found by CodeQL in scoped module.

5.6 Nancy

Analysis Output Sample:



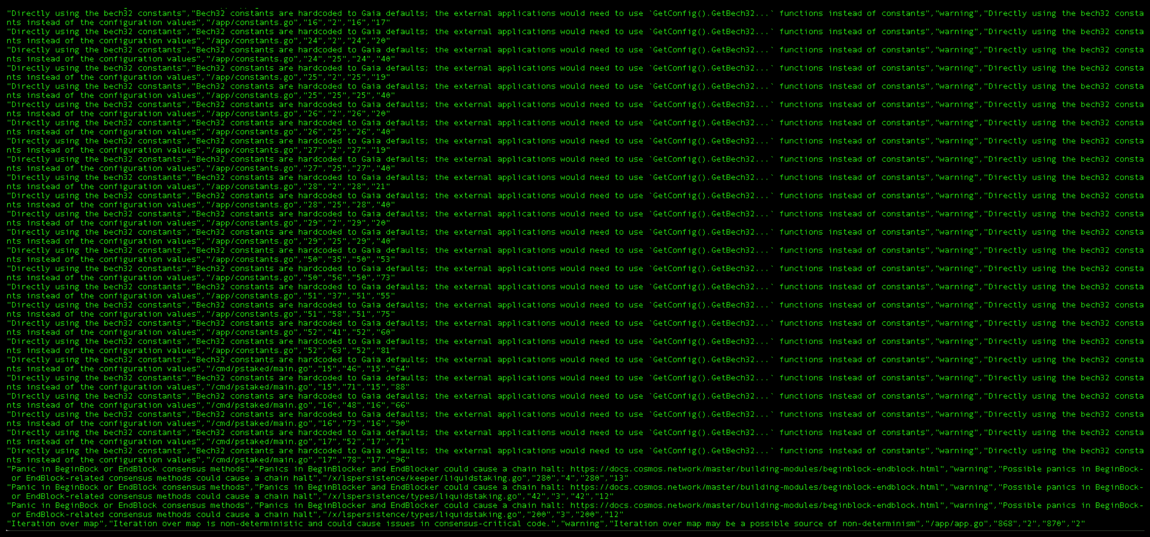
```
pkg:golang/golang.org/x/net@v0.12.0
1 known vulnerabilities affecting installed version
| [CVE-2023-3978] CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') |
| Description        | Text nodes not in the HTML namespace are incorrectly literally rendered,
|                    | causing text which should be escaped to not be. This could lead to an XSS
|                    | attack.
| OSS Index ID       | CVE-2023-3978
| CVSS Score         | 6.1/10 (Medium)
| CVSS Vector        | CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
| Link for more info | https://ossindex.sonatype.org/vulnerability/CVE-2023-3978?component-type=golang&component-name=golang.org%2Fx%2Fnet&utm_source=nancy-client&utm_medium=integration&utm_content=0.0.0-dev

1 Vulnerable Packages

| Summary                    |
| Audited Dependencies | 142 |
| Vulnerable Dependencies | 1 |
```

Figure 3: Nancy results

- No major issues found by Nancy.

5.7 Codeql

Analysis Output Sample:



```
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","16","2","16","17"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","24","2","24","20"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","24","25","24","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","25","2","25","19"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","25","25","25","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","26","2","26","20"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","26","25","26","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","27","2","27","19"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","27","25","27","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","28","2","28","21"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","28","25","28","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","29","2","29","20"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","29","25","29","40"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","50","35","50","53"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","50","56","50","73"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","51","37","51","55"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","51","58","51","75"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","52","41","52","60"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/app/constants.go","52","63","52","81"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","15","46","15","64"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","15","71","15","88"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","16","48","16","66"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","16","73","16","90"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","17","52","17","71"
"Directly using the bech32 constants","Bech32 constants are hardcoded to Gaia defaults; the external applications would need to use `GetConfig().GetBech32...` functions instead of constants","warning","Directly using the bech32 consta
nts instead of the configuration values","/cmd/pstaked/main.go","17","78","17","96"
"Panic in BeginBock or EndBlock consensus methods","Panics in BeginBlocker and EndBlocker could cause a chain halt: https://docs.cosmos.network/master/building-modules/beginblock-endblock.html","warning","Possible panics in BeginBock-
 or EndBlock-related consensus methods could cause a chain halt","/x/lspersistence/keeper/liquidstaking.go","280","4","280","13"
"Panic in BeginBock or EndBlock consensus methods","Panics in BeginBlocker and EndBlocker could cause a chain halt: https://docs.cosmos.network/master/building-modules/beginblock-endblock.html","warning","Possible panics in BeginBock-
 or EndBlock-related consensus methods could cause a chain halt","/x/lspersistence/types/liquidstaking.go","42","3","42","12"
"Panic in BeginBock or EndBlock consensus methods","Panics in BeginBlocker and EndBlocker could cause a chain halt: https://docs.cosmos.network/master/building-modules/beginblock-endblock.html","warning","Possible panics in BeginBock-
 or EndBlock-related consensus methods could cause a chain halt","/x/lspersistence/types/liquidstaking.go","200","3","200","12"
"Iteration over map","Iteration over map is non-deterministic and could cause issues in consensus-critical code.","warning","Iteration over map may be a possible source of non-determinism","/app/app.go","868","2","870","2"
```

Figure 4: Nancy results

- No major issues found by Codeql.

THANK YOU FOR CHOOSING

HALBORN