HALBORN

Composable Finance - IBC-Go Module

Golang Security Audit

DRAFT

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	02/22/2023	Gokberk Gulgun
0.2	Document Updates	02/28/2023	Gokberk Gulgun
0.3	Draft Review	02/28/2023	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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Composable Finance is a modular and interoperable platform for decentralized finance (DeFi) that is designed for developers and end-users who want to create new solutions across multiple blockchain infrastructures. Their technology stack is like lego blocks, allowing developers to add new logic and create novel solutions that can be deployed across different blockchain networks. The platform also aims to explore the ability to segment different aspects of a protocol into different layers, so that they can take advantage of specific benefits while still being able to natively communicate through Composable's infrastructure.

Composable Finance engaged Halborn to conduct a security audit on the ibc-go module, beginning on February 15th, 2023 and ending on February 28th, 2023 .

1.2 AUDIT SUMMARY

The team at Halborn was provided one week for the engagement and assigned a full-time security engineer to audit the security of the dynamic inflation feature. The security engineer is a blockchain and smart contract security expert with advanced penetration testing, smart contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that the **IBC Go** operates as intended.
- Identify potential security issues with the custom modules used in the IBC Go.

In summary, Halborn identified some improvements to reduce the likelihood and impact of risks, which should be addressed by the Composable Finance 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 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 audit:

- 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 **IBC Go**.

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.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

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

1.4 SCOPE

1. IN-SCOPE TREE & COMMIT :

- URL: /strangelove-ventures/ibc-go.

- Commit hash: d0c9da3694668330c1f04bf9c82e994491894a3f.

The review was scoped to the ibc-go directory in the strangelove-ventures repository.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	3	3	6

LIKELIHOOD

IMPACT

	(HAL-01)			
(HAL-04) (HAL-05) (HAL-06)		(HAL-02) (HAL-03)		
(HAL-10)				
(HAL-07) (HAL-08) (HAL-09) (HAL-11) (HAL-12)				

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - CONSENSUS STATE IS NOT SET IN THE WASM LIGHTCLIENT	Medium	-
HAL-02 - WASMVM PARAMETERS ARE NOT SET IN THE CONTRACT INTERACTIONS	Medium	-
HAL-03 - THE STATUS OF CLIENT STATE IS MARKED AS ACTIVE BY DEFAULT	Medium	-
HAL-04 - SIGNER IS NOT VALIDATED IN THE MSGPUSHNEWWASMCODE'S VALIDATEBASIC FUNCTION	Low	-
HAL-05 - GRPC-GATEWAY ROUTES ARE NOT REGISTERED	Low	-
HAL-06 - TEST DOCKER IMAGE RUNNING AS ROOT	Low	-
HAL-07 - OUT OF DATE GOLANG VERSION	Low	-
HAL-08 - CHANNEL DEFINITION IS MISSING ON THE MSGCHANNELOPENTRYRESPONSE	Informational	-
HAL-09 - LACK OF CONDITIONAL CHECK WHEN BLOCK HEIGHT - BLOCK TIMESTAMP EQUALS TO 0	Informational	-
HAL-10 - OPEN TO-DOS	Informational	-
HAL-11 - GOLANG LIBRARY IS IMPORTED MULTIPLE TIMES	Informational	-
HAL-12 - MATH RAND IS NOT SAFE FOR CONCURRENT USE	Informational	-

FINDINGS & TECH DETAILS

3.1 (HAL-01) CONSENSUS STATE IS NOT SET IN THE WASM LIGHTCLIENT - MEDIUM

Description:

The IBC clients are lightweight clients that operate on the blockchain. Each of these clients is distinguished by a unique client-id, which is automatically generated by appending the client type and the global client counter in the format of client-type-N. The clients keep track of consensus states on other blockchains, including the proof spec required to validate proofs against the client's consensus state. Clients can have multiple connections to the counterparty chain. In the context of a WebAssembly (WASM) lightclient, the consensus state is not being properly set, which can lead to inconsistencies in the state of the network and potential security vulnerabilities. The failure to set the consensus state in the WASM lightclient can result in a range of negative impacts on the blockchain network. For example, it can lead to forked chains, where different nodes have different views of the network's state, and transactions may be processed incorrectly or not at all.

Code Location:

/modules/light-clients/08-wasm/client_state.go#L242

```
Listing 1

1 func (c ClientState) UpdateState(ctx sdk.Context, cdc codec.
↳ BinaryCodec, clientStore sdk.KVStore, clientMsg exported.
↳ ClientMessage) []exported.Height {
2     const VerifyClientMessage = "update_state"
3     inner := make(map[string]interface{})
4     inner["client_state"] = c
5     clientMsgConcrete := make(map[string]interface{})
6     switch clientMsg := clientMsg.(type) {
7     case *Header:
8         clientMsgConcrete["header"] = clientMsg
```

```
 9      case *Misbehaviour:
10          clientMsgConcrete["misbehaviour"] = clientMsg
11      }
12      inner["client_message"] = clientMsgConcrete
13      payload := make(map[string]map[string]interface{})
14      payload[VerifyClientMessage] = inner
15
16      output, err := call[contractResult](payload, &c, ctx,
↳ clientStore)
17      if err != nil {
18          panic(err)
19      }
20      if err := json.Unmarshal(output.Data, &c); err != nil {
21          panic(sdkerrors.Wrapf(ErrUnableToUnmarshalPayload, fmt.
↳ Sprintf("underlying error: %s", err.Error())))
22      }
23
24      setClientState(clientStore, cdc, &c)
25      // TODO: do we need to set consensus state?
26      // setConsensusState(clientStore, cdc, consensusState, header.
↳ GetHeight())
27      // setConsensusMetadata(ctx, clientStore, header.GetHeight())
28
29      return []exported.Height{c.LatestHeight}
30 }
```

Risk Level:

Likelihood - 2
Impact - 4

Recommendation:

To address this issue, the Composable Finance team should ensure that the consensus state is properly set in the WASM lightclient. This may involve implementing additional checks or procedures to verify the consensus states of other blockchains.

3.2 (HAL-02) WASMVM PARAMETERS ARE NOT SET IN THE CONTRACT INTERACTIONS - MEDIUM

Description:

During the code review, It has been noticed that some variables are not set in the functions of the WASMVM interactions. For instance, The system experiences an issue where the Querier is not defined on the func (*VM) Instantiate function. This can occur when the system attempts to access the Querier object before it is initialized or when the object is not defined in the code. As a result, the system is unable to perform certain operations that require access to the Querier object, which can lead to errors and malfunctions.

Code Location:

/modules/light-clients/08-wasm/vm.go#L119

```
Listing 2
1 func initContract(codeID []byte, ctx sdk.Context, store sdk.
↳ KVStore) (*types.Response, error) {
2     gasMeter := ctx.GasMeter()
3     chainID := ctx.BlockHeader().ChainID
4     height := ctx.BlockHeader().Height
5     // safety checks before casting below
6     if height < 0 {
7         panic("Block height must never be negative")
8     }
9     sec := ctx.BlockTime().Unix()
10    if sec < 0 {
11        panic("Block (unix) time must never be negative ")
12    }
13    env := types.Env{
14        Block: types.BlockInfo{
15            Height:  uint64(height),
16            Time:    uint64(sec),
```

```
17                ChainID: chainID,
18            },
19            Contract: types.ContractInfo{
20                Address: "",
21            },
22        }
23
24        msgInfo := types.MessageInfo{
25            Sender: "",
26            Funds:  nil,
27        }
28        // mockFailureAPI := *api.NewMockFailureAPI()
29        // mockQuerier := api.MockQuerier{}
30
31        desercost := types.UFraction{Numerator: 0, Denominator: 1}
32        response, _, err := WasmVM.Instantiate(codeID, env, msgInfo,
   []byte("{}"), store, cosmwasm.GoAPI{}, nil, gasMeter, gasMeter.
   Limit(), desercost)
33        return response, err
34 }
```

/modules/light-clients/08-wasm/vm.go#L162

```
Listing 3

 1 func callContractWithEnvAndMeter(codeID cosmwasm.Checksum, ctx sdk
   .Context, store sdk.KVStore, env types.Env, gasMeter sdk.GasMeter,
    msg []byte) (*types.Response, error) {
 2     msgInfo := types.MessageInfo{
 3         Sender: "",
 4         Funds:  nil,
 5     }
 6     // TODO: fix this
 7     // mockFailureAPI := *api.NewMockFailureAPI()
 8     // mockQuerier := api.MockQuerier{}
 9     desercost := types.UFraction{Numerator: 1, Denominator: 1}
10     resp, gasUsed, err := WasmVM.Execute(codeID, env, msgInfo, msg
   , store, cosmwasm.GoAPI{}, nil, gasMeter, gasMeter.Limit(),
   desercost)
11     if &ctx != nil {
12         consumeGas(ctx, gasUsed)
13     }
14     return resp, err
```

```
15 }
```

Listing 4

```go
1 func queryContractWithStore(codeID cosmwasm.Checksum, store sdk.
↳ KVStore, msg []byte) ([]byte, error) {
2     // TODO: fix this
3     // mockEnv := api.MockEnv()
4     // mockGasMeter := api.NewMockGasMeter(1)
5     // mockFailureAPI := *api.NewMockFailureAPI()
6     // mockQuerier := api.MockQuerier{}
7     // TODO: figure out what this is for
8     desercost := types.UFraction{Numerator: 1, Denominator: 1}
9     resp, _, err := WasmVM.Query(codeID, types.Env{}, msg, store,
↳ cosmwasm.GoAPI{}, nil, nil, maxGasLimit, desercost)
10    return resp, err
11 }
```

Undefined Parameters:

- Querier
- GO API
- MsgInfo
- Gas Meter
- Env

Listing 5

```go
1 func (vm *VM) Execute(
2     checksum Checksum,
3     env types.Env,
4     info types.MessageInfo,
5     executeMsg []byte,
6     store KVStore,
7     goapi GoAPI,
8     querier Querier,
9     gasMeter GasMeter,
10    gasLimit uint64,
```

```
11      deserCost types.UFraction,
12 ) (*types.Response, uint64, error)
13
14 func (vm *VM) Instantiate(
15     checksum Checksum,
16     env types.Env,
17     info types.MessageInfo,
18     initMsg []byte,
19     store KVStore,
20     goapi GoAPI,
21     querier Querier,
22     gasMeter GasMeter,
23     gasLimit uint64,
24     deserCost types.UFraction,
25 ) (*types.Response, uint64, error)
26
27 func (vm *VM) Query(
28     checksum Checksum,
29     env types.Env,
30     queryMsg []byte,
31     store KVStore,
32     goapi GoAPI,
33     querier Querier,
34     gasMeter GasMeter,
35     gasLimit uint64,
36     deserCost types.UFraction,
37 ) ([]byte, uint64, error)
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

To address this issue, it is recommended to review the system's code
to ensure that all the objects are properly defined and initialized.
This may involve adding appropriate checks and error handling mechanisms
to prevent the system from accessing the relative objects before it is
defined.

3.3 (HAL-03) THE STATUS OF CLIENT STATE IS MARKED AS ACTIVE BY DEFAULT - MEDIUM

Description:

The IBC clients are lightweight clients that operate on the blockchain. Each of these clients is distinguished by a unique client-id, which is automatically generated by appending the client type and the global client counter in the format of client-type-N. The clients keep track of consensus states on other blockchains, including the proof spec required to validate proofs against the client's consensus state. Clients can have multiple connections to the counterparty chain.

The ClientState contains information specific to the blockchain and the lightweight client that is necessary to verify updates and upgrades to the IBC client. This information can include chain-id, latest height, proof specs, unbonding periods, or the status of the lightweight client. The client state in a lightweight client is marked as Active by default, which means that it is assumed to be functioning properly and able to receive and process transactions. However, this assumption can be dangerous as it can lead to the network being vulnerable to attacks or errors that may occur in the client state.

Code Location:

/modules/light-clients/08-wasm/client_state.go#L38

```
Listing 6
1 func (c ClientState) Status(ctx sdk.Context, store sdk.KVStore,
↳ cdc codec.BinaryCodec) exported.Status {
2     return exported.Active
3 }
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

It is recommended to implement a more robust approach to the client state status.

FINDINGS & TECH DETAILS

3.4 (HAL-04) SIGNER IS NOT VALIDATED IN THE MSGPUSHNEWWASMCODE'S VALIDATEBASIC FUNCTION - LOW

Description:

The validate basic function is responsible for checking the basic validity of a transaction message, including verifying that the signer address is valid and that the transaction payload conforms to the expected format. However, the validate basic function does not properly validate the signer address.

Code Location:

/modules/light-clients/08-wasm/msgs.go#L12

```
Listing 7
1 package wasm
2
3 import (
4     sdk "github.com/cosmos/cosmos-sdk/types"
5     sdkerrors "github.com/cosmos/cosmos-sdk/types/errors"
6 )
7
8 var (
9     _ sdk.Msg = &MsgPushNewWasmCode{}
10 )
11
12 func (m MsgPushNewWasmCode) ValidateBasic() error {
13     if len(m.Code) == 0 {
14         return sdkerrors.Wrapf(ErrWasmEmptyCode,
15             "empty wasm code",
16         )
17     }
18
19     return nil
20 }
21
```

```
22
23  func (m MsgPushNewWasmCode) GetSigners() []sdk.AccAddress {
24      signer, err := sdk.AccAddressFromBech32(m.Signer)
25      if err != nil {
26          panic(err)
27      }
28      return []sdk.AccAddress{signer}
29  }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

To address this issue, the validate basic function should include additional checks to verify the authenticity of the signer address, such as checking that the address corresponds to a valid public key.

3.5 (HAL-05) GRPC-GATEWAY ROUTES ARE NOT REGISTERED - LOW

Description:

The HTTP handlers in the module are responsible for handling incoming queries and requests from other nodes in the network. However, if the HTTP handlers are not properly registered for the service query to mux, all gRPC queries will fail. This can prevent the network from processing transactions, retrieving data, or executing other functions.

Code Location:

/modules/light-clients/08-wasm/module.go#L47

```
Listing 8

1 func (AppModuleBasic) RegisterGRPCGatewayRoutes(clientCtx client.
↳ Context, mux *runtime.ServeMux) {}
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

To address this issue, it is recommended to review the HTTP handlers carefully in the wasm module and ensure that they are properly registered for the service query to mux.

3.6 (HAL-06) TEST DOCKER IMAGE RUNNING AS ROOT - LOW

Description:

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

Code Location:

Dockerfile

```
Listing 9

1  FROM golang:1.19 as builder
2
3  ARG IBC_GO_VERSION
4
5  ENV GOPATH=""
6  ENV GOMODULE="on"
7
8  # ensure the ibc go version is being specified for this image.
9  RUN test -n "${IBC_GO_VERSION}"
10
11 COPY go.mod .
12 COPY go.sum .
13
14 RUN go mod download
15
16 ADD internal internal
17 ADD testing testing
18 ADD modules modules
19 ADD LICENSE LICENSE
20
21 COPY contrib/devtools/Makefile contrib/devtools/Makefile
22 COPY Makefile .
23
```

```
24
25 RUN make build
26
27 FROM ubuntu:20.04
28
29 ARG IBC_GO_VERSION
30
31 LABEL "org.cosmos.ibc-go" "${IBC_GO_VERSION}"
32
33 COPY --from=builder /go/build/simd /bin/simd
34
35 ENTRYPOINT ["simd"]
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

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

```
Listing 10:  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.
```

3.7 (HAL-07) OUT OF DATE GOLANG VERSION - INFORMATIONAL

Description:

Go 1.20 was released about a week ago, and for those of us in the blockchain community, there are some important notes to take note of go1.20. Specifically, the update includes performance improvements in garbage collection by up to 2%, stronger cryptography operations against side-channel attacks due to constant-time cryptography, and a warning about the math/big package, which is often used heavily in blockchain projects. It's also important to remember that Go's support policy only includes the two newest major releases, which means that as of now, 1.19 and 1.20 are supported, while 1.18 has been deprecated.

Code Location:

/go.mod#L1

```
Listing 11

1 go 1.19
2
3 module github.com/cosmos/ibc-go/v7
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Consider upgrading the go version.

3.8 (HAL-08) CHANNEL DEFINITION IS MISSING ON THE MSGCHANNELOPENTRYRESPONSE - INFORMATIONAL

Description:

The MsgChannelOpenTryResponse is a message type used in the Inter-Blockchain Communication (IBC) protocol to handle the response to a ChannelOpenTry message. In a recent version of ibc-go, a new parameter, channel id, has been added to this message type. This parameter is useful because it provides a unique identifier for the channel, which is required for further interactions between the two blockchains involved in the communication.

Code Location:

/modules/core/keeper/msg_server.go#L277

```
Listing 12

 1 unc (k Keeper) ChannelOpenTry(goCtx context.Context, msg *
↳ channeltypes.MsgChannelOpenTry) (*channeltypes.
↳ MsgChannelOpenTryResponse, error) {
 2     ctx := sdk.UnwrapSDKContext(goCtx)
 3
 4     // Lookup module by port capability
 5     module, portCap, err := k.PortKeeper.LookupModuleByPort(ctx,
↳ msg.PortId)
 6     if err != nil {
 7         ctx.Logger().Error("channel open try callback failed", "
↳ port-id", msg.PortId, "error", sdkerrors.Wrap(err, "could not
↳ retrieve module from port-id"))
 8         return nil, sdkerrors.Wrap(err, "could not retrieve module
↳  from port-id")
 9     }
10
11     // Retrieve application callbacks from router
```

```
12      cbs, ok := k.Router.GetRoute(module)
13      if !ok {
14          ctx.Logger().Error("channel open try callback failed", "
↳ port-id", msg.PortId, "error", sdkerrors.Wrapf(porttypes.
↳ ErrInvalidRoute, "route not found to module: %s", module))
15          return nil, sdkerrors.Wrapf(porttypes.ErrInvalidRoute, "
↳ route not found to module: %s", module)
16      }
17
18      // Perform 04-channel verification
19      channelID, cap, err := k.ChannelKeeper.ChanOpenTry(ctx, msg.
↳ Channel.Ordering, msg.Channel.ConnectionHops, msg.PortId,
20          portCap, msg.Channel.Counterparty, msg.CounterpartyVersion
↳ , msg.ProofInit, msg.ProofHeight,
21      )
22      if err != nil {
23          ctx.Logger().Error("channel open try callback failed", "
↳ error", sdkerrors.Wrap(err, "channel handshake open try failed"))
24          return nil, sdkerrors.Wrap(err, "channel handshake open
↳ try failed")
25      }
26
27      // Perform application logic callback
28      version, err := cbs.OnChanOpenTry(ctx, msg.Channel.Ordering,
↳ msg.Channel.ConnectionHops, msg.PortId, channelID, cap, msg.
↳ Channel.Counterparty, msg.CounterpartyVersion)
29      if err != nil {
30          ctx.Logger().Error("channel open try callback failed", "
↳ port-id", msg.PortId, "channel-id", channelID, "error", err.Error
↳ ())
31          return nil, sdkerrors.Wrapf(err, "channel open try
↳ callback failed for port ID: %s, channel ID: %s", msg.PortId,
↳ channelID)
32      }
33
34      // Write channel into state
35      k.ChannelKeeper.WriteOpenTryChannel(ctx, msg.PortId, channelID
↳ , msg.Channel.Ordering, msg.Channel.ConnectionHops, msg.Channel.
↳ Counterparty, version)
36
37      ctx.Logger().Info("channel open try callback succeeded", "
↳ channel-id", channelID, "port-id", msg.PortId, "version", version)
38
39      return &channeltypes.MsgChannelOpenTryResponse{
```

```
40         Version: version,
41     }, nil
42 }
```

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to include the new channel id parameter in the MsgChannelOpenTryResponse message type. Additionally, they should take advantage of this parameter to improve the security and usability of their IBC implementations. It is also recommended to update any relevant documentation or communication protocols to ensure that all parties involved in the IBC communication are aware of the new parameter and its importance.

FINDINGS & TECH DETAILS

3.9 (HAL-09) LACK OF CONDITIONAL CHECK WHEN BLOCK HEIGHT - BLOCK TIMESTAMP EQUALS TO 0 - INFORMATIONAL

Description:

During the testing, It has been noticed that The VM module lack a conditional check when the block height and block timestamp equal to zero.

Code Location:

/light-clients/08-wasm/vm.go#L93-L97

```
Listing 13

 1 func initContract(codeID []byte, ctx sdk.Context, store sdk.
 ↳ KVStore) (*types.Response, error) {
 2     gasMeter := ctx.GasMeter()
 3     chainID := ctx.BlockHeader().ChainID
 4     height := ctx.BlockHeader().Height
 5     // safety checks before casting below
 6     if height < 0 {
 7         panic("Block height must never be negative")
 8     }
 9     sec := ctx.BlockTime().Unix()
10     if sec < 0 {
11         panic("Block (unix) time must never be negative ")
12     }
13 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

To address this issue, it is recommended to modify the system's code to handle the scenario when the block height and block timestamp are equal to 0. This can involve adding appropriate checks and error handling mechanisms to prevent unexpected behavior on the vm.

DRAFT

3.10 (HAL-10) OPEN TO-DOS - INFORMATIONAL

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:

```
Listing 14

 1 ./proto/buf.yaml:4:  # TODO: update sdk buf dependency when v0
↳ .47.0 is tagged and pushed to BSR
 2 ./docs/DOCS_README.md:92:TODO: update or remove
 3 ./docs/ibc/light-clients/setup.md:101:- TODO: update when params
↳ are managed by ibc-go
 4 ./docs/architecture/adr-015-ibc-packet-receiver.md:253:    // TODO
↳ : Source chain sent invalid packet, shutdown channel
 5 ./docs/.vuepress/config.js:56:    // TODO
 6 ./testing/simapp/sim_test.go:277:// TODO: Make another test for
↳ the fuzzer itself, which just has noOp txs
 7 ./testing/simapp/config.go:23:  FlagOnOperationValue      bool
↳ // TODO: Remove in favor of binary search for invariant violation
 8 ./testing/simapp/app.go:921:    // TODO: update git tag in link
↳ below
 9 ./testing/simapp/state.go:101:      // TODO(fdymylja/jonathan):
↳ should we panic in this case
10 ./testing/wasm.go:89:   // TODO: Wasm.CreateMisbehaviour
11 ./testing/endpoint.go:100:     // TODO
12 ./testing/endpoint.go:169:// TODO: implement actual upgrade chain
↳ functionality via scheduling an upgrade
13 ./modules/light-clients/08-wasm/client_state.go:37:// TODO call
↳ into the contract here to get the status once it is implemented in
↳  the contract, for now, returns active
14 ./modules/light-clients/08-wasm/client_state.go:241:    // TODO:
↳ do we need to set consensus state?
```

```
15 ./modules/light-clients/08-wasm/vm.go:13:// TODO: figure out
↳ better handling for the gas settings. ideally these should be in
↳ the
16 ./modules/light-clients/08-wasm/vm.go:56:// TODO: Move this into
↳ the 28-wasm keeper
17 ./modules/light-clients/08-wasm/vm.go:87:// TODO: Move this into a
↳  public method on the 28-wasm keeper
18 ./modules/light-clients/08-wasm/vm.go:124:// TODO: Move this into
↳ a public method on the 28-wasm keeper
19 ./modules/light-clients/08-wasm/vm.go:152:// TODO: Move this into
↳ a private method on the 28-wasm keeper
20 ./modules/light-clients/08-wasm/vm.go:158:   // TODO: fix this
21 ./modules/light-clients/08-wasm/vm.go:169:// TODO: Move this into
↳ a public method on the 28-wasm keeper
22 ./modules/light-clients/08-wasm/vm.go:171:   // TODO: fix this
23 ./modules/light-clients/08-wasm/vm.go:176:   // TODO: figure out
↳ what this is for
24 ./modules/light-clients/08-wasm/wasm_test.go:51:    // TODO:
↳ deprecate usage in favor of testing package
25 ./modules/light-clients/08-wasm/wasm_test.go:321:// TODO:
↳ uncomment when test data is aquired
26 ./modules/light-clients/08-wasm/wasm_test.go:381:// TODO:
↳ uncomment when fisherman is merged
27 ./modules/light-clients/08-wasm/wasm_test.go:429:// TODO:
↳ uncomment when test data is aquired
28 ./modules/light-clients/07-tendermint/tendermint_test.go:46:    //
↳  TODO: deprecate usage in favor of testing package
29 ./modules/light-clients/07-tendermint/tendermint_test.go:67:    //
↳  TODO: deprecate usage in favor of testing package
30 ./modules/core/02-client/keeper/keeper_test.go:71:  // TODO:
↳ deprecate
31 ./modules/core/02-client/keeper/keeper_test.go:127: // TODO:
↳ deprecate
32 ./modules/core/04-channel/client/cli/cli.go:32:     // TODO: next
↳ sequence Send ?
33 ./modules/core/keeper/keeper_test.go:35:    // TODO: remove
34 ./modules/core/module.go:122:   // TODO:
35 ./modules/apps/transfer/keeper/mbt_relay_test.go:160:
↳ bankAfter:   BalancesFromTla(tc.BankAfter), // TODO different
↳ semantics
36 ./modules/apps/transfer/module.go:99:   // TODO
37 ./e2e/go.mod:208:// TODO: using version v1.0.0 causes a build
↳ failure. This is the previous version which compiles successfully.
```

Risk Level:

Likelihood - 1
Impact - 2

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.

FINDINGS & TECH DETAILS

3.11 (HAL-11) GOLANG LIBRARY IS IMPORTED MULTIPLE TIMES - INFORMATIONAL

Description:

When a Golang library is imported multiple times in a codebase, it leads to redundant code and can result in larger binary sizes. In some cases, it can also cause conflicts between different versions of the same library that may have been imported.

Code Location:

/modules/light-clients/08-wasm/client_state.go#L8-L9

Listing 15

```
1
2 import (
3 ...
4     "github.com/cosmos/cosmos-sdk/types"
5     sdk "github.com/cosmos/cosmos-sdk/types"
6 ...
7 )
8
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to avoid importing the same library multiple times.

3.12 (HAL-12) MATH RAND IS NOT SAFE FOR CONCURRENT USE - INFORMATIONAL

Description:

Using math/rand is deterministic random by default, and it is dangerous. During the code review, It has been noticed that several code locations uses math/rand function.

Code Location:

```
Listing 16

 1 ./modules/core/module.go:7: "math/rand"
 2 ./modules/apps/transfer/simulation/genesis_test.go:5:   "math/rand
↳ "
 3 ./modules/apps/transfer/simulation/params.go:5: "math/rand"
 4 ./modules/apps/transfer/simulation/params_test.go:4:   "math/rand
↳ "
 5 ./modules/apps/transfer/simulation/genesis.go:6:    "math/rand"
 6 ./modules/apps/transfer/module.go:7:     "math/rand"
 7 ./modules/apps/27-interchain-accounts/simulation/genesis_test.go
↳ :5: "math/rand"
 8 ./modules/apps/27-interchain-accounts/simulation/params.go:5:   "
↳ math/rand"
 9 ./modules/apps/27-interchain-accounts/simulation/params_test.go:5:
↳    "math/rand"
10 ./modules/apps/27-interchain-accounts/simulation/genesis.go:6:   "
↳ math/rand"
11 ./modules/apps/27-interchain-accounts/module.go:7:   "math/rand"
12 ./modules/apps/29-fee/module.go:7:    "math/rand"
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use crypto/rand instead of math/rand.

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**, **codeql** and **nancy**. After Halborn verified all the code 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.

AUTOMATED TESTING

Gosec - Security Analysis Output Sample:

Listing 17

```
1
2 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/testing/simapp/
  test_helpers.go:216] - G404 (CWE-338): Use of weak random number
  generator (math/rand instead of crypto/rand) (Confidence: MEDIUM,
  Severity: HIGH)
3     215:     tx, err := simtestutil.GenSignedMockTx(
4   > 216:         rand.New(rand.NewSource(time.Now().UnixNano())),
5     217:         txCfg,
6
7
8
9 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/testing/simapp/
  helpers/test_helpers.go:27] - G404 (CWE-338): Use of weak random
  number generator (math/rand instead of crypto/rand) (Confidence:
  MEDIUM, Severity: HIGH)
10    26:     // create a random length memo
11  > 27:     r := rand.New(rand.NewSource(time.Now().UnixNano()))
12    28:
13
14
15
16 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/testing/simapp/
  state.go:187] - G304 (CWE-22): Potential file inclusion via
  variable (Confidence: HIGH, Severity: MEDIUM)
17    186: func AppStateFromGenesisFileFn(r io.Reader, cdc codec.
  JSONCodec, genesisFile string) (tmtypes.GenesisDoc, []simtypes.
  Account) {
18  > 187:     bytes, err := os.ReadFile(genesisFile)
19    188:     if err != nil {
20
21
22
23 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/light-
  clients/08-wasm/cli/tx.go:27] - G304 (CWE-22): Potential file
  inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
24    26:
25  > 27:         code, err := ioutil.ReadFile(fileName)
26    28:         if err != nil {
27
28
29
```

```
30 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/03-
↳ connection/client/utils/utils.go:207] - G304 (CWE-22): Potential
↳ file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
31    206:          // check for file path if JSON input is not
↳ provided
32  > 207:             contents, err := os.ReadFile(arg)
33    208:             if err != nil {
34
35
36
37 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/03-
↳ connection/client/utils/utils.go:189] - G304 (CWE-22): Potential
↳ file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
38    188:          // check for file path if JSON input is not
↳ provided
39  > 189:             contents, err := os.ReadFile(arg)
40    190:             if err != nil {
41
42
43
44 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/03-
↳ connection/client/utils/utils.go:172] - G304 (CWE-22): Potential
↳ file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
45    171:          // check for file path if JSON input is not
↳ provided
46  > 172:             contents, err := os.ReadFile(arg)
47    173:             if err != nil {
48
49
50
51 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:357] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
52    356:                // check for file path if JSON input is
↳ not provided
53  > 357:                contents, err := os.ReadFile(
↳ clientContentOrFileName)
54    358:                if err != nil {
55
56
57
58 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:217] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
```

```
59      216:                 // check for file path if JSON input is
↳ not provided
60   > 217:                 contents, err := os.ReadFile(
↳ consensusContentOrFileName)
61      218:                 if err != nil {
62
63
64
65 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:201] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
66      200:                 // check for file path if JSON input is
↳ not provided
67   > 201:                 contents, err := os.ReadFile(
↳ clientContentOrFileName)
68      202:                 if err != nil {
69
70
71
72 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:154] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
73      153:                 // check for file path if JSON input is
↳ not provided
74   > 154:                 contents, err := os.ReadFile(
↳ misbehaviourContentOrFileName)
75      155:                 if err != nil {
76
77
78
79 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:107] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
80      106:                 // check for file path if JSON input is
↳ not provided
81   > 107:                 contents, err := os.ReadFile(
↳ clientMsgContentOrFileName)
82      108:                 if err != nil {
83
84
85
86 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:62] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
```

```
 87      61:                 // check for file path if JSON input is
↳ not provided
 88    > 62:                 contents, err := os.ReadFile(
↳ consensusContentOrFileName)
 89      63:                 if err != nil {
 90
 91
 92
 93 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/core/02-
↳ client/client/cli/tx.go:46] - G304 (CWE-22): Potential file
↳ inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
 94      45:                 // check for file path if JSON input is
↳ not provided
 95    > 46:                 contents, err := os.ReadFile(
↳ clientContentOrFileName)
 96      47:                 if err != nil {
 97
 98
 99
100 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/modules/apps/27-
↳ interchain-accounts/controller/client/cli/tx.go:84] - G304 (CWE
↳ -22): Potential file inclusion via variable (Confidence: HIGH,
↳ Severity: MEDIUM)
101      83:                 // check for file path if JSON input is
↳ not provided
102    > 84:                 contents, err := os.ReadFile(
↳ msgContentOrFileName)
103      85:                 if err != nil {
104
105
106
107 [ibc-go-d0c9da3694668330c1f04bf9c82e994491894a3f/testing/simapp/
↳ export.go:180] - G104 (CWE-703): Errors unhandled. (Confidence:
↳ HIGH, Severity: LOW)
108      179:
109    > 180:     iter.Close()
110      181:
111
112
113
114 Summary:
115   Gosec : 2.15.0
116   Files : 363
117   Lines : 110117
```

```
118    Nosec  : 0
119    Issues : 16
```

Semgrep - Security Analysis Output Sample:

Listing 18

```
1  Scanning across multiple languages:
2      <multilang> |  52 rules Œ 1334 files
3               go |  86 rules Œ  364 files
4             json |   4 rules Œ   67 files
5             yaml |  27 rules Œ   18 files
6             bash |   4 rules Œ    5 files
7           python | 244 rules Œ    1 file
8               js | 179 rules Œ    1 file
9       dockerfile |   4 rules Œ    1 file
10
11   100%||1791/1791  tasks
12
13
14
15   Results
16
17
18  Findings:
19
20    modules/apps/27-interchain-accounts/controller/types/query.pb.gw
↳  .go
21       trailofbits.go.questionable-assignment.questionable-
↳  assignment
22          Should `protoReq` be modified when an error could be
↳  returned?
23          Details: https://sg.run/qq6y
24
25           52 protoReq.Owner, err = runtime.String(val)
26             ----------------------------------------
27           63 protoReq.ConnectionId, err = runtime.String(val)
28             ----------------------------------------
29           90 protoReq.Owner, err = runtime.String(val)
30             ----------------------------------------
31          101 protoReq.ConnectionId, err = runtime.String(val)
32
```

```
33
34    modules/apps/29-fee/types/query.pb.gw.go
35        trailofbits.go.questionable-assignment.questionable-
↳ assignment
36          Should `protoReq` be modified when an error could be
↳ returned?
37          Details: https://sg.run/qq6y
38
39          208 protoReq.ChannelId, err = runtime.String(val)
40              ------------------------------------------
41          219 protoReq.PortId, err = runtime.String(val)
42              ------------------------------------------
43          253 protoReq.ChannelId, err = runtime.String(val)
44              ------------------------------------------
45          264 protoReq.PortId, err = runtime.String(val)
46              ------------------------------------------
47          646 protoReq.ChannelId, err = runtime.String(val)
48              ------------------------------------------
49          657 protoReq.Relayer, err = runtime.String(val)
50              ------------------------------------------
51          684 protoReq.ChannelId, err = runtime.String(val)
52              ------------------------------------------
53          695 protoReq.Relayer, err = runtime.String(val)
54              ------------------------------------------
55          722 protoReq.ChannelId, err = runtime.String(val)
56              ------------------------------------------
57          733 protoReq.Relayer, err = runtime.String(val)
58              ------------------------------------------
59          760 protoReq.ChannelId, err = runtime.String(val)
60              ------------------------------------------
61          771 protoReq.Relayer, err = runtime.String(val)
62              ------------------------------------------
63          834 protoReq.ChannelId, err = runtime.String(val)
64              ------------------------------------------
65          845 protoReq.PortId, err = runtime.String(val)
66              ------------------------------------------
67          872 protoReq.ChannelId, err = runtime.String(val)
68              ------------------------------------------
69          883 protoReq.PortId, err = runtime.String(val)
70
71
72    modules/apps/transfer/keeper/grpc_query.go
73        trailofbits.go.invalid-usage-of-modified-variable.invalid-
↳ usage-of-modified-variable
```

```
74          Variable `hash` is likely modified and later used on error
└ . In some cases this could result
75          in panics due to a nil dereference
76          Details: https://sg.run/WWQ2
77
78       26 hash, err := types.ParseHexHash(strings.TrimPrefix(req
└ .Hash, "ibc/"))
79          27 if err != nil {
80          28     return nil, status.Error(codes.InvalidArgument, fmt
└ .Sprintf("invalid denom trace hash: %s, error: %s", hash.String(),
└  err))
81          29 }
82
83
84   modules/apps/transfer/types/query.pb.gw.go
85      trailofbits.go.questionable-assignment.questionable-
└ assignment
86          Should `protoReq` be modified when an error could be
└ returned?
87          Details: https://sg.run/qq6y
88
89       52 protoReq.Hash, err = runtime.String(val)
90          ---------------------------------------
91       79 protoReq.Hash, err = runtime.String(val)
92          ---------------------------------------
93      160 protoReq.Trace, err = runtime.String(val)
94          ---------------------------------------
95      187 protoReq.Trace, err = runtime.String(val)
96          ---------------------------------------
97      214 protoReq.ChannelId, err = runtime.String(val)
98          ---------------------------------------
99      225 protoReq.PortId, err = runtime.String(val)
100         ---------------------------------------
101     252 protoReq.ChannelId, err = runtime.String(val)
102         ---------------------------------------
103     263 protoReq.PortId, err = runtime.String(val)
104
105
106  modules/core/02-client/types/query.pb.gw.go
107     trailofbits.go.questionable-assignment.questionable-
└ assignment
108         Should `protoReq` be modified when an error could be
└ returned?
109         Details: https://sg.run/qq6y
```

47

```
110
111        52 protoReq.ClientId, err = runtime.String(val)
112           --------------------------------------
113        79 protoReq.ClientId, err = runtime.String(val)
114           --------------------------------------
115       146 protoReq.ClientId, err = runtime.String(val)
116           --------------------------------------
117       157 protoReq.RevisionNumber, err = runtime.Uint64(val)
118           --------------------------------------
119       168 protoReq.RevisionHeight, err = runtime.Uint64(val)
120           --------------------------------------
121       202 protoReq.ClientId, err = runtime.String(val)
122           --------------------------------------
123       213 protoReq.RevisionNumber, err = runtime.Uint64(val)
124           --------------------------------------
125       224 protoReq.RevisionHeight, err = runtime.Uint64(val)
126           --------------------------------------
127       262 protoReq.ClientId, err = runtime.String(val)
128           --------------------------------------
129       296 protoReq.ClientId, err = runtime.String(val)
130           --------------------------------------
131       334 protoReq.ClientId, err = runtime.String(val)
132           --------------------------------------
133       368 protoReq.ClientId, err = runtime.String(val)
134           --------------------------------------
135       402 protoReq.ClientId, err = runtime.String(val)
136           --------------------------------------
137       429 protoReq.ClientId, err = runtime.String(val)
138
139
140   modules/core/03-connection/types/query.pb.gw.go
141      trailofbits.go.questionable-assignment.questionable-
 ↳ assignment
142        Should `protoReq` be modified when an error could be
 ↳ returned?
143        Details: https://sg.run/qq6y
144
145        52 protoReq.ConnectionId, err = runtime.String(val)
146           --------------------------------------
147        79 protoReq.ConnectionId, err = runtime.String(val)
148           --------------------------------------
149       142 protoReq.ClientId, err = runtime.String(val)
150           --------------------------------------
151       169 protoReq.ClientId, err = runtime.String(val)
```

```
152            ------------------------------------------
153        196 protoReq.ConnectionId, err = runtime.String(val)
154            ------------------------------------------
155        223 protoReq.ConnectionId, err = runtime.String(val)
156            ------------------------------------------
157        250 protoReq.ConnectionId, err = runtime.String(val)
158            ------------------------------------------
159        261 protoReq.RevisionNumber, err = runtime.Uint64(val)
160            ------------------------------------------
161        272 protoReq.RevisionHeight, err = runtime.Uint64(val)
162            ------------------------------------------
163        299 protoReq.ConnectionId, err = runtime.String(val)
164            ------------------------------------------
165        310 protoReq.RevisionNumber, err = runtime.Uint64(val)
166            ------------------------------------------
167        321 protoReq.RevisionHeight, err = runtime.Uint64(val)
168
169
170  modules/core/04-channel/types/query.pb.gw.go
171      trailofbits.go.questionable-assignment.questionable-
  ↳ assignment
172        Should `protoReq` be modified when an error could be
  ↳ returned?
173        Details: https://sg.run/qq6y
174
175         52 protoReq.ChannelId, err = runtime.String(val)
176            ------------------------------------------
177         63 protoReq.PortId, err = runtime.String(val)
178            ------------------------------------------
179         90 protoReq.ChannelId, err = runtime.String(val)
180            ------------------------------------------
181        101 protoReq.PortId, err = runtime.String(val)
182            ------------------------------------------
183        168 protoReq.Connection, err = runtime.String(val)
184            ------------------------------------------
185        202 protoReq.Connection, err = runtime.String(val)
186            ------------------------------------------
187        236 protoReq.ChannelId, err = runtime.String(val)
188            ------------------------------------------
189        247 protoReq.PortId, err = runtime.String(val)
190            ------------------------------------------
191        274 protoReq.ChannelId, err = runtime.String(val)
192            ------------------------------------------
193        285 protoReq.PortId, err = runtime.String(val)
```

```
194             ----------------------------------------
195         312 protoReq.ChannelId, err = runtime.String(val)
196             ----------------------------------------
197         323 protoReq.PortId, err = runtime.String(val)
198             ----------------------------------------
199         334 protoReq.RevisionNumber, err = runtime.Uint64(val)
200             ----------------------------------------
201         345 protoReq.RevisionHeight, err = runtime.Uint64(val)
202             ----------------------------------------
203         372 protoReq.ChannelId, err = runtime.String(val)
204             ----------------------------------------
205         383 protoReq.PortId, err = runtime.String(val)
206             ----------------------------------------
207         394 protoReq.RevisionNumber, err = runtime.Uint64(val)
208             ----------------------------------------
209         405 protoReq.RevisionHeight, err = runtime.Uint64(val)
210             ----------------------------------------
211         432 protoReq.ChannelId, err = runtime.String(val)
212             ----------------------------------------
213         443 protoReq.PortId, err = runtime.String(val)
214             ----------------------------------------
215         454 protoReq.Sequence, err = runtime.Uint64(val)
216             ----------------------------------------
217         481 protoReq.ChannelId, err = runtime.String(val)
218             ----------------------------------------
219         492 protoReq.PortId, err = runtime.String(val)
220             ----------------------------------------
221         503 protoReq.Sequence, err = runtime.Uint64(val)
222             ----------------------------------------
223         534 protoReq.ChannelId, err = runtime.String(val)
224             ----------------------------------------
225         545 protoReq.PortId, err = runtime.String(val)
226             ----------------------------------------
227         579 protoReq.ChannelId, err = runtime.String(val)
228             ----------------------------------------
229         590 protoReq.PortId, err = runtime.String(val)
230             ----------------------------------------
231         624 protoReq.ChannelId, err = runtime.String(val)
232             ----------------------------------------
233         635 protoReq.PortId, err = runtime.String(val)
234             ----------------------------------------
235         646 protoReq.Sequence, err = runtime.Uint64(val)
236             ----------------------------------------
237         673 protoReq.ChannelId, err = runtime.String(val)
```

```
238          ----------------------------------------
239      684 protoReq.PortId, err = runtime.String(val)
240          ----------------------------------------
241      695 protoReq.Sequence, err = runtime.Uint64(val)
242          ----------------------------------------
243      722 protoReq.ChannelId, err = runtime.String(val)
244          ----------------------------------------
245      733 protoReq.PortId, err = runtime.String(val)
246          ----------------------------------------
247      744 protoReq.Sequence, err = runtime.Uint64(val)
248          ----------------------------------------
249      771 protoReq.ChannelId, err = runtime.String(val)
250          ----------------------------------------
251      782 protoReq.PortId, err = runtime.String(val)
252          ----------------------------------------
253      793 protoReq.Sequence, err = runtime.Uint64(val)
254          ----------------------------------------
255      824 protoReq.ChannelId, err = runtime.String(val)
256          ----------------------------------------
257      835 protoReq.PortId, err = runtime.String(val)
258          ----------------------------------------
259      869 protoReq.ChannelId, err = runtime.String(val)
260          ----------------------------------------
261      880 protoReq.PortId, err = runtime.String(val)
262          ----------------------------------------
263      914 protoReq.ChannelId, err = runtime.String(val)
264          ----------------------------------------
265      925 protoReq.PortId, err = runtime.String(val)
266          ----------------------------------------
267      936 protoReq.PacketCommitmentSequences, err = runtime.
       ↳ Uint64Slice(val, ",")
268          ----------------------------------------
269      963 protoReq.ChannelId, err = runtime.String(val)
270          ----------------------------------------
271      974 protoReq.PortId, err = runtime.String(val)
272          ----------------------------------------
273      985 protoReq.PacketCommitmentSequences, err = runtime.
       ↳ Uint64Slice(val, ",")
274          ----------------------------------------
275     1012 protoReq.ChannelId, err = runtime.String(val)
276          ----------------------------------------
277     1023 protoReq.PortId, err = runtime.String(val)
278          ----------------------------------------
```

```
279        1034 protoReq.PacketAckSequences, err = runtime.Uint64Slice
  ↳ (val, ",")
280             ------------------------------------------
281        1061 protoReq.ChannelId, err = runtime.String(val)
282             ------------------------------------------
283        1072 protoReq.PortId, err = runtime.String(val)
284             ------------------------------------------
285        1083 protoReq.PacketAckSequences, err = runtime.Uint64Slice
  ↳ (val, ",")
286             ------------------------------------------
287        1110 protoReq.ChannelId, err = runtime.String(val)
288             ------------------------------------------
289        1121 protoReq.PortId, err = runtime.String(val)
290             ------------------------------------------
291        1148 protoReq.ChannelId, err = runtime.String(val)
292             ------------------------------------------
293        1159 protoReq.PortId, err = runtime.String(val)
294
295
296   modules/light-clients/08-wasm/query.pb.gw.go
297       trailofbits.go.questionable-assignment.questionable-
  ↳ assignment
298         Should `protoReq` be modified when an error could be
  ↳ returned?
299         Details: https://sg.run/qq6y
300
301          52 protoReq.CodeId, err = runtime.String(val)
302             ------------------------------------------
303          79 protoReq.CodeId, err = runtime.String(val)
304
305
306   testing/endpoint.go
307       trailofbits.go.questionable-assignment.questionable-
  ↳ assignment
308         Should `endpoint` be modified when an error could be
  ↳ returned?
309         Details: https://sg.run/qq6y
310
311         131 endpoint.ClientID, err = ParseClientIDFromEvents(res.
  ↳ GetEvents())
312             ------------------------------------------
313         233 endpoint.ConnectionID, err =
  ↳ ParseConnectionIDFromEvents(res.GetEvents())
314             ------------------------------------------
```

```
315          259 endpoint.ConnectionID, err =
 ↳ ParseConnectionIDFromEvents(res.GetEvents())
316          ----------------------------------------
317          341 endpoint.ChannelID, err = ParseChannelIDFromEvents(res
 ↳ .GetEvents())
318          ----------------------------------------
319          372 endpoint.ChannelID, err = ParseChannelIDFromEvents(res
 ↳ .GetEvents())
```

AUTOMATED TESTING

THANK YOU FOR CHOOSING

// HALBORN