Tabi

BLOCKCHAIN AUDIT REPORT

January 2025

ExVul

www.exvul.com

Table of Contents

1. EXECUTIVE SUMMARY

Exvul Web3 Security was engaged by go-helios to review Blockchain implementation. The assessment was conducted in accordance with our systematic approach to evaluate potential security issues based upon customer requirement. The report provides detailed recommendations to resolve the issue and provide additional suggestions or recommendations for improvement.

The outcome of the assessment outlined in chapter 3 provides the system's owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the underlying technical issue.

1.1 Methodology

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology [10] which is the gold standard in risk assessment using the following risk models:

- Likelihood: represents how likely a particular vulnerability is to be uncovered and exploited in the wild.
- Impact: measures the technical loss and business damage of a successful attack.
- Severity: determine the overall criticality of the risk.

Likelihood can be: High, Medium and Low and impact are categorized into for: High, Medium, Low, Informational. Severity is determined by likelihood and impact and can be classified into five categories accordingly, Critical, High, Medium, Low, Informational shown in table 1.1.

Likelihood \ IMPACT	Informational	Low	Medium	High
High	Informational	Medium	High	Critical
Medium	Informational	Low	Medium	High
Low	Informational	Low	Low	Medium

Table 1.1 Overall Risk Severity

To evaluate the risk, we will be going through a list of items, and each would be labelled with a severity category. The audit was performed with a systematic approach guided by a comprehensive assessment list carefully designed to identify known and impactful security issues. If our tool or analysis does not identify any issue, the contract can be considered safe regarding the assessed item. For any discovered issue, we might further deploy contracts on our private test environment and run tests to confirm the findings. If necessary, we would additionally build a PoC to demonstrate the possibility of exploitation. The concrete list of check items is shown in Table 1.2.

- Basic Coding Bugs: We first statically analyze given Blockchain with our proprietary static code analyzer for known coding bugs, and then manually verify (reject or confirm) all the issues found by our tool.
- Code and business security testing: We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs.
- Additional Recommendations: We also provide additional suggestions regarding the coding and development of Blockchains from the perspective of proven programming practices.

Category	Assessment Item
P2P Communication Security	Connection Number Occupation Audit
	Eclipse Attack
	Packet Size Limit
	Node Communication Protocol Security
RPC Interface Security	RPC Sensitive Interface Permissions
	Traditional Web Security
	RPC Interface Security
Consensus Mechanism Security	Design Of Consensus Mechanism
	Implementation Of Consensus Verification
	Incentive Mechanism Audit
Transaction processing Security	Transaction Signature Logic
	Transaction Verification Logic
	Transaction Processing Logic
	Transaction Fee Setting
	Transaction Replay
Cryptography Security	Random Number Range And Probability Distribution
	Cryptographic Algorithm Lmplementation/Use
Wallet Module & Account Security Audit	Private Key / Mnemonic Word Storage Security
	Private Key / Mnemonic Word Usage Security
	Private key/mnemonic generation algorithm
Others Security Audit	Database Security
	Thread Security
	File Permission Security
	Historical Vulnerability Security

Table 1.2: The Full List of Assessment Items

To better describe each issue we identified, we categorize the findings with Common Weakness Enumeration (CWE-699) [14], which is a community-developed list of software weakness types to better delineate and organize weaknesses around concepts frequently encountered in software development.

2. FINDINGS OVERVIEW

2.1 Project Info

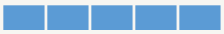
Project Name: Tabi

Audit Time: December 24, 2024 – February 7, 2025

Language: go-lang

File Name	Link
Electra	https://github.com/tabilabs/tabi
Commit Hash	44e732be7ae81630ffc9b27faa51351cebd624c7

2.2 Summary

Severity	Found	
Critical	7	
High	0	
Medium	1	
Low	5	
Informational	1	

2.3 Key Findings

ID	Severity	Findings Title	Status	Confirm
NVE-001	Critical	Nil Transaction Panic in Ethereum Transaction Handling	Fixed	Confirmed
NVE-002	Low	Missing Chain ID in LegacyTx Type	Acknowledged	Confirmed
NVE-003	Low	Missing Handling for LegacyTx Type	Fixed	Confirmed
NVE-004	Low	Missing Validation in Transaction Creation Functions	Fixed	Confirmed
NVE-005	Critical	Integer Overflow exists in the function	Fixed	Confirmed
NVE-006	Critical	Integer Underflow exists in the function	Fixed	Confirmed
NVE-007	Critical	Integer Overflow exists in the function UpdateNode	Fixed	Confirmed
NVE-008	Critical	TotalCount Validation Missing	Fixed	Confirmed
NVE-009	Critical	AuthorizedMembers Cannot Be Fully Removed	Fixed	Confirmed
NVE-010	Low	Inefficient Looping in AuthorizedMembers Deletion	Fixed	Confirmed
NVE-011	Low	Inefficient Looping in AuthorizedMembers Addition	Fixed	Confirmed
NVE-012	Medium	Precision Loss in int64 to float32 Conversion	Acknowledged	Confirmed
NVE-013	Critical	Missing nil Check for msg.AsTransaction() Result	Fixed	Confirmed
NVE-014	Informational	Acknowledged Static scan result	Acknowledged	Confirmed

Table 2.3: Key Audit Findings

3. DETAILED DESCRIPTION OF FINDINGS

3.1 Nil Transaction Panic in Ethereum Transaction Handling

ID:	NVE-001	**Location:**	rpc/backend/utils.go
Severity:	Critical	**Category:**	Transaction processing Security
Likelihood:	Medium	**Impact:**	High

Description:

The code lacks proper nil-check handling when calling the **AsTransaction** method on **ethMsg**. If **ethMsg.AsTransaction()** returns nil, the subsequent call to **tx.EffectiveGasTipValue(blockBaseFee)** will cause the program to panic due to dereferencing a nil pointer. This could lead to a node crash and compromise network reliability.

```
   for i := 0; i < tendermintTxCount; i++ {
        eachTendermintTx := tendermintTxs[i]
        eachTendermintTxResult := tendermintTxResults[i]

        tx, err := b.clientCtx.TxConfig.TxDecoder()(eachTendermintTx)
        if err != nil {
            b.logger.Debug("failed to decode transaction in block", "heigh
t", blockHeight, "error", err.Error())
            continue
        }
        txGasUsed := uint64(eachTendermintTxResult.GasUsed) // #nosec G70
1
        for _, msg := range tx.GetMsgs() {
            ethMsg, ok := msg.(*evmtypes.MsgEthereumTx)
            if !ok {
                continue
            }
            tx := ethMsg.AsTransaction()
            reward := tx.EffectiveGasTipValue(blockBaseFee)
            if reward == nil {
                reward = big.NewInt(0)
            }
            sorter = append(sorter, types.TxGasAndReward{GasUsed: txGasUse
d, Reward: reward})
        }
    }
```

If an error occurs, **AsTransaction** will return nil.

```
   // AsTransaction creates an Ethereum Transaction type from the msg fie
lds
func (msg MsgEthereumTx) AsTransaction() *ethtypes.Transaction {
    txData, err := UnpackTxData(msg.Data)
    if err != nil {
        return nil
    }

    return ethtypes.NewTx(txData.AsEthereumData())
}
```

Result: Confirmed

Fix Result: Fixed

3.2 Missing Chain ID in LegacyTx Type

ID:	NVE-002	**Location:**	x/evm/types/msg.go
Severity:	Low	**Category:**	Transaction processing Security
Likelihood:	Medium	**Impact:**	High

Description:

The **LegacyTx** type, does not include a ChainID. This issue arises because the ChainID is often required for transaction validation, signing, and anti-replay protection in Ethereum and Ethereum-compatible ecosystems.

```
    switch {
    case tx.Accesses == nil:
        txData = &LegacyTx{
            To:       toAddr,
            Amount:   amt,
            GasPrice: gp,
            Nonce:    tx.Nonce,
            GasLimit: tx.GasLimit,
            Data:     tx.Input,
        }
    case tx.Accesses != nil && tx.GasFeeCap != nil && tx.GasTipCap != nil:

        gtc := sdkmath.NewIntFromBigInt(tx.GasTipCap)
        gfc := sdkmath.NewIntFromBigInt(tx.GasFeeCap)

        txData = &DynamicFeeTx{
            ChainID:   cid,
            Amount:    amt,
            To:        toAddr,
            GasTipCap: &gtc,
            GasFeeCap: &gfc,
            Nonce:     tx.Nonce,
            GasLimit:  tx.GasLimit,
            Data:      tx.Input,
            Accesses:  NewAccessList(tx.Accesses),
        }
```

Result: Confirmed

Fix Result: Acknowledged

3.3 Missing Handling for LegacyTx Type

ID:	NVE-003	Location:	x/evm/types/tx_data.go
Severity:	Low	Category:	Transaction processing Security
Likelihood:	Medium	Impact:	Medium

Description:

In the **NewTxDataFromTx** function, the default case is currently used to handle transactions of type **ethtypes.LegacyTx.** This general handling approach can lead to unintended consequences.

```
// NOTE: All non-protected transactions (i.e non EIP155 signed) will fail
 if the
// AllowUnprotectedTxs parameter is disabled.
func NewTxDataFromTx(tx *ethtypes.Transaction) (TxData, error) {
    var txData TxData
    var err error
    switch tx.Type() {
    case ethtypes.DynamicFeeTxType:
        txData, err = NewDynamicFeeTx(tx)
    case ethtypes.AccessListTxType:
        txData, err = newAccessListTx(tx)
    default:
        txData, err = NewLegacyTx(tx)
    }
    if err != nil {
        return nil, err
    }

    return txData, nil
}
```

Result: Confirmed

Fix Result: Fixed

3.4 Missing Validation in Transaction Creation Functions

ID:	NVE-004	Location:	x/evm/types/legacy_tx.go
Severity:	Low	Category:	Transaction processing Security
Likelihood:	Medium	Impact:	High

Description:

In the **NewLegacyTx NewDynamicTx newAccessListTx** should add validate function of **tx** data.

```
func NewLegacyTx(tx *ethtypes.Transaction) (*LegacyTx, error) {
    txData := &LegacyTx{
        Nonce:    tx.Nonce(),
        Data:     tx.Data(),
        GasLimit: tx.Gas(),
    }

    v, r, s := tx.RawSignatureValues()
    if to := tx.To(); to != nil {
        txData.To = to.Hex()
    }

    if tx.Value() != nil {
        amountInt, err := types.SafeNewIntFromBigInt(tx.Value())
        if err != nil {
            return nil, err
        }
        txData.Amount = &amountInt
    }

    if tx.GasPrice() != nil {
        gasPriceInt, err := types.SafeNewIntFromBigInt(tx.GasPrice())
        if err != nil {
            return nil, err
        }
        txData.GasPrice = &gasPriceInt
    }

    txData.SetSignatureValues(tx.ChainId(), v, r, s)
    return txData, nil
}
```

Result: Confirmed

Fix Result: Fixed

3.5 Integer Overflow exists in the function CommitComputingPower

ID:	NVE-005	**Location:**	x/captains/keeper/members.go
Severity:	Critical	**Category:**	Transaction processing Security
Likelihood:	Medium	**Impact:**	Medium

Description:

The **CommitComputingPower** function does not check for overflow when adding `amount` to **`before`**. This could lead to incorrect results if the sum exceeds the maximum value for **`uint64`**.

```
// CommitComputingPower commits the pending computing power.
func (k Keeper) CommitComputingPower(ctx sdk.Context, amount uint64, owne
r sdk.AccAddress) (uint64, uint64) {
    before := k.GetClaimableComputingPower(ctx, owner)
    after := before + amount
    k.setClaimableComputingPower(ctx, after, owner)
    return before, after
}
```

Result: Confirmed

Fix Result: Fixed

3.6 Integer Underflow exists in the function decrClaimbleComputingPower

ID:	NVE-006	Location:	x/captains/keeper/computing_power.go
Severity:	Critical	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Medium

Description:

The **decrClaimableComputingPower** function does not check if the `amount` to be decremented is greater than the current power. This can lead to an underflow and incorrect state.

```
// decrClaimableComputingPower decrements the claimable computing power o
f an owner.
func (k Keeper) decrClaimableComputingPower(ctx sdk.Context, amount uint6
4, owner sdk.AccAddress) {
    power := k.GetClaimableComputingPower(ctx, owner)
    power -= amount
    k.setClaimableComputingPower(ctx, power, owner)
}
```

Result: Confirmed

Fix Result: Fixed

3.7 Integer Overflow exists in the function UpdateNode

ID:	NVE-007	Location:	x/captains/keeper/nodes.go
Severity:	Critical	Category:	Transaction processing Security
Likelihood:	Medium	Impact:	Medium

Description:

The function **UpdateNode** is responsible for modifying the computing power associated with a node. It lacks a validation to handle potential integer overflow in the computation of **after := node.ComputingPower + amount.**

```
// UpdateNode defines a method for updating the computing power of the sp
ecified node
func (k Keeper) UpdateNode(
    ctx sdk.Context,
    nodeID string,
    amount uint64,
    owner sdk.AccAddress,
) error {
    node, found := k.GetNode(ctx, nodeID)
    if !found {
        return errorsmod.Wrap(types.ErrNodeNotExists, nodeID)
    }

    if err := k.AuthorizeNode(ctx, nodeID, owner); err != nil {
        return err
    }

    claimable := k.GetClaimableComputingPower(ctx, owner)
    if claimable < amount {
        return errorsmod.Wrap(types.ErrInsufficientComputingPower, nodeI
D)
    }

    after := node.ComputingPower + amount
    currDivision, _ := k.GetDivision(ctx, node.DivisionId)
    if after > currDivision.ComputingPowerUpperBound {
        // check if we need to improve node division
        nextDivision := k.DecideDivision(ctx, after)
        node.DivisionId = nextDivision.Id
        k.incrDivisionTotalCount(ctx, nextDivision)
        k.decrDivisionTotalCount(ctx, currDivision)
    }

    // set node info
    node.ComputingPower = after
    if err := k.setNode(ctx, node); err != nil {
        return err
    }
```

Result: Confimed

Fix Result: Fixed

3.8 TotalCount Validation Missing

ID:	NVE-008	**Location:**	x/captains/keeper/division.go
Severity:	Critical	**Category:**	Transaction processing Security
Likelihood:	Medium	**Impact:**	Low

Description:

Should check **TotalCount** is smaller than zero or not.

```
// decrDivisionTotalCount decrements the sold count of the division
func (k Keeper) decrDivisionTotalCount(ctx sdk.Context, division types.Di
vision) {
    division.TotalCount--
    k.setDivision(ctx, division)
}
```

Result: Confirmed

Fix Result: Fixed

3.9 AuthorizedMembers Cannot Be Fully Removed

ID:	NVE-009	Location:	x/captains/keeper/members.go
Severity:	Critical	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

The **DeleteAuthorizedMembers** function does not validate whether the **AuthorizedMembers** array should always retain at least one member. If all members are deleted without replacement, no authorized operations will be functional.

```
// DeleteAuthorizedMembers deletes the list of authorized members
func (k Keeper) DeleteAuthorizedMembers(ctx sdk.Context, members []strin
g) error {
    params := k.GetParams(ctx)

    events := make([]sdk.Event, 0)
    for _, member := range members {
        allowRemove := false
        for i, authzMember := range params.AuthorizedMembers {
            if authzMember == member {
                params.AuthorizedMembers = append(params.AuthorizedMembers
[:i], params.AuthorizedMembers[i+1:]...)
                allowRemove = true
            }
        }
        if allowRemove {
            events = append(
                events,
                sdk.NewEvent(
                    types.EventTypeRemoveAuthorizedMembers,
                    sdk.NewAttribute(types.AttributeKeyAuthorizedMember, me
mber),
                ),
            )
        }
    }
```

Result: Confirmed

Fix Result: Fixed

3.10 Inefficient Looping in AuthorizedMembers Deletion

ID:	NVE-010	Location:	x/captains/keeper/members.go
Severity:	Low	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

If the **AuthorizedMembers** list contains the specified member and if **allowRemove** is set to true, we should terminate the internal loop.

```
// DeleteAuthorizedMembers deletes the list of authorized members
func (k Keeper) DeleteAuthorizedMembers(ctx sdk.Context, members []strin
g) error {
    params := k.GetParams(ctx)

    events := make([]sdk.Event, 0)
    for _, member := range members {
        allowRemove := false
        for i, authzMember := range params.AuthorizedMembers {
            if authzMember == member {
                params.AuthorizedMembers = append(params.AuthorizedMembers
[:i], params.AuthorizedMembers[i+1:]...)
                allowRemove = true
            }
        }
        if allowRemove {
            events = append(
                events,
                sdk.NewEvent(
                    types.EventTypeRemoveAuthorizedMembers,
                    sdk.NewAttribute(types.AttributeKeyAuthorizedMember, me
mber),
                ),
            )
        }
    }
```

Result: Confirmed

Fix Result: Fixed

3.11 Inefficient Looping in AuthorizedMembers Addition

ID:	NVE-011	Location:	x/captains/keeper/members.go
Severity:	Low	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

If found the **AuthorizedMembers** and **allowAdd** set to true, we should break the internal loop.

```
// SetAuthorizedMembers sets the list of authorized members
func (k Keeper) SetAuthorizedMembers(ctx sdk.Context, members []string) e
rror {
    params := k.GetParams(ctx)
    events := make([]sdk.Event, 0)

    for _, member := range members {
        allowAdd := true
        for _, authzMember := range params.AuthorizedMembers {
            if authzMember == member {
                allowAdd = false
            }
        }
        if allowAdd {
            params.AuthorizedMembers = append(params.AuthorizedMembers, me
mber)
            events = append(
                events,
                sdk.NewEvent(
                    types.EventTypeAddAuthorizedMembers,
                    sdk.NewAttribute(types.AttributeKeyAuthorizedMember, me
mber),
                ),
            )
        }
    }
```

Result: Confirmed

Fix Result: Fixed

3.12 Precision Loss in int64 to float32 Conversion

ID:	NVE-012	Location:	go.mod
Severity:	Medium	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

Converting an **int64** to a **float32** can lead to precision loss because `float32` has a limited precision compared to `int64`. If the values of **a.Amount.Int64()** are large, this conversion might not be safe.

```
    defer func() {
        for _, a := range amount {
            if a.Amount.IsInt64() {
                telemetry.SetGaugeWithLabels(
                    []string{"tx", "msg", "claims"},
                    float32(a.Amount.Int64()),
                    []metrics.Label{telemetry.NewLabel("denom", a.Denom)},
                )
            }
        }
    }()
```

Result: Confirmed

Fix Result: Acknowledged

3.13 Missing nil Check for msg.AsTransaction Result

ID:	NVE-013	Location:	go.mod
Severity:	Critical	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

It is necessary to verify whether the return value of **tx := msg.AsTransaction()** is nil. If the return value is nil, it may cause the node to crash.

```
// parameter.
func (k *Keeper) EthereumTx(goCtx context.Context, msg *types.MsgEthereum
Tx) (*types.MsgEthereumTxResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    sender := msg.From
    tx := msg.AsTransaction()
    txIndex := k.GetTxIndexTransient(ctx)
```

Result: Confirmed

Fix Result: Fixed

3.14 Static scan result

ID:	NVE-014	Location:	go.mod
Severity:	Informational	Category:	Transaction processing Security
Likelihood:	Low	Impact:	Low

Description:

We used the **govulncheck** tool to scan the dependencies used in the project and found many significant issues, including critical vulnerabilities in the outdated Cosmos SDK.

```
Vulnerability #1: GO-2025-3420
    Sensitive headers incorrectly sent after cross-domain redirect in net/
http
  More info: https://pkg.go.dev/vuln/GO-2025-3420
  Standard library
    Found in: net/http@go1.23.4
    Fixed in: net/http@go1.23.5
    Example traces found:
      #1: rpc/websockets.go:319:24: rpc.websocketsServer.tcpGetAndSendResp
onse calls http.Client.Do
      #2: x/evm/types/query.pb.go:1560:20: types.queryClient.BaseFee calls
 grpc.ClientConn.Invoke, which eventually calls http.Client.PostForm
      #3: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls http.Get

Vulnerability #2: GO-2025-3373
    Usage of IPv6 zone IDs can bypass URI name constraints in crypto/x509
  More info: https://pkg.go.dev/vuln/GO-2025-3373
  Standard library
    Found in: crypto/x509@go1.23.4
    Fixed in: crypto/x509@go1.23.5
    Example traces found:
      #1: testutil/network/network.go:627:15: network.Network.Cleanup call
s grpc.Server.Stop, which eventually calls x509.CertPool.AppendCertsFromP
EM
      #2: server/json_rpc.go:88:26: server.StartJSONRPC calls http.Server.
Serve, which eventually calls x509.Certificate.Verify
      #3: server/util.go:77:37: server.ConnectTmWS calls client.WSClient.O
nStart, which eventually calls x509.Certificate.VerifyHostname
      #4: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTender
mintBlock calls x509.HostnameError.Error
      #5: testutil/network/network.go:627:15: network.Network.Cleanup call
s grpc.Server.Stop, which eventually calls x509.ParseCertificate
      #6: rpc/websockets.go:102:32: rpc.Start calls http.ListenAndServeTL
S, which eventually calls x509.ParseECPrivateKey
      #7: rpc/websockets.go:102:32: rpc.Start calls http.ListenAndServeTL
S, which eventually calls x509.ParsePKCS1PrivateKey
      #8: rpc/websockets.go:102:32: rpc.Start calls http.ListenAndServeTL
```

```
S, which eventually calls x509.ParsePKCS8PrivateKey
      #9: testutil/network/network.go:627:15: network.Network.Cleanup call
s grpc.Server.Stop, which eventually calls x509.ParsePKIXPublicKey

Vulnerability #3: GO-2024-3339
    Transaction decoding may result in a stack overflow or resource exhaus
tion
    in github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-3339
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.15
    Example traces found:
      #1: x/evm/types/msg.go:321:27: types.MsgEthereumTx.UnpackInterfaces
calls types.interfaceRegistry.UnpackAny
      #2: rpc/backend/blocks.go:431:46: backend.Backend.EthMsgsFromTenderm
intBlock calls tx.DefaultTxDecoder, which calls unknownproto.RejectUnknow
nFields
      #3: rpc/backend/blocks.go:431:46: backend.Backend.EthMsgsFromTenderm
intBlock calls tx.DefaultTxDecoder, which calls unknownproto.RejectUnknow
nFieldsStrict

Vulnerability #4: GO-2024-3279
    Mismatched bit-length validation in can lead to panic in cosmossdk.io/
math
  More info: https://pkg.go.dev/vuln/GO-2024-3279
  Module: cosmossdk.io/math
    Found in: cosmossdk.io/math@v1.0.0-rc.0
    Fixed in: cosmossdk.io/math@v1.4.0
    Example traces found:
      #1: x/feemarket/types/params.go:6:2: types.init calls math.init, whi
ch eventually calls math.LegacyDec.Quo

Vulnerability #5: GO-2024-2948
    Code Execution on Git update in github.com/hashicorp/go-getter
  More info: https://pkg.go.dev/vuln/GO-2024-2948
  Module: github.com/hashicorp/go-getter
    Found in: github.com/hashicorp/go-getter@v1.7.0
    Fixed in: github.com/hashicorp/go-getter@v1.7.5
    Example traces found:
      #1: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls getter.Get
      #2: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls getter.GetFile

Vulnerability #6: GO-2024-2874
    Inter-Blockchain Communication (IBC) protocol "Huckleberry" vulnerabil
ity in
    github.com/cosmos/ibc-go
  More info: https://pkg.go.dev/vuln/GO-2024-2874
  Module: github.com/cosmos/ibc-go/v6
    Found in: github.com/cosmos/ibc-go/v6@v6.1.1
    Fixed in: N/A
```

```
    Example traces found:
      #1: app/app.go:618:26: app.Tabi.BeginBlocker calls module.Manager.Be
ginBlock, which eventually calls 02.BeginBlocker
      #2: app/export.go:44:34: app.Tabi.ExportAppStateAndValidators calls
module.Manager.ExportGenesis, which eventually calls 02.ExportGenesis
      #3: cmd/tabid/root.go:171:35: tabid.queryCommand calls module.BasicM
anager.AddQueryCommands, which eventually calls 02.GetQueryCmd
      #4: cmd/tabid/root.go:198:32: tabid.txCommand calls module.BasicMana
ger.AddTxCommands, which eventually calls 02.GetTxCmd
      #5: app/app.go:649:27: app.Tabi.InitChainer calls module.Manager.Ini
tGenesis, which eventually calls 02.InitGenesis
      ...
      ...
      #3289: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which
eventually calls utils.QueryTendermintHeader
      #3290: app/app.go:77:2: app.init calls transfer.init, which eventual
ly calls utils.init
      #3291: app/modules.go:48:2: app.init calls client.init, which eventu
ally calls utils.init
      #3292: app/modules.go:47:2: app.init calls core.init, which eventual
ly calls utils.init

Vulnerability #7: GO-2024-2800
    Argument injection when fetching remote default Git branches in
    github.com/hashicorp/go-getter
  More info: https://pkg.go.dev/vuln/GO-2024-2800
  Module: github.com/hashicorp/go-getter
    Found in: github.com/hashicorp/go-getter@v1.7.0
    Fixed in: github.com/hashicorp/go-getter@v1.7.4
    Example traces found:
      #1: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls getter.Get
      #2: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls getter.GetFile

Vulnerability #8: GO-2024-2694
    Potential Reentrancy using Timeout Callbacks in ibc-hooks in
    github.com/cosmos/ibc-go
  More info: https://pkg.go.dev/vuln/GO-2024-2694
  Module: github.com/cosmos/ibc-go/v6
    Found in: github.com/cosmos/ibc-go/v6@v6.1.1
    Fixed in: github.com/cosmos/ibc-go/v6@v6.3.0
    Example traces found:
      #1: app/ante/evm/eth.go:377:13: evm.EthIncrementSenderSequenceDecora
tor.AnteHandle calls types.ChainAnteDecorators, which eventually calls ke
eper.Keeper.Timeout
      #2: app/ante/evm/eth.go:377:13: evm.EthIncrementSenderSequenceDecora
tor.AnteHandle calls types.ChainAnteDecorators, which eventually calls ke
eper.Keeper.TimeoutOnClose

Vulnerability #9: GO-2024-2687
    HTTP/2 CONTINUATION flood in net/http
  More info: https://pkg.go.dev/vuln/GO-2024-2687
```

```
  Module: golang.org/x/net
    Found in: golang.org/x/net@v0.9.0
    Fixed in: golang.org/x/net@v0.23.0
    Example traces found:
      #1: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which eve
ntually calls http2.ConfigureTransports
      #2: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTender
mintBlock calls http2.ConnectionError.Error
      #3: testutil/network/network.go:638:26: network.Network.Cleanup call
s log.tmLogger.Error, which eventually calls http2.ErrCode.String
      #4: testutil/network/network.go:638:26: network.Network.Cleanup call
s log.tmLogger.Error, which eventually calls http2.FrameHeader.String
      #5: testutil/network/network.go:638:26: network.Network.Cleanup call
s log.tmLogger.Error, which eventually calls http2.FrameType.String
      #6: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls http2.Framer.ReadFrame
      #7: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls http2.Framer.WriteContinuation
      #8: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls http2.Framer.WriteData
      #9: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls http2.Framer.WriteGoAway
      #10: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WriteHeaders
      #11: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WritePing
      #12: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WriteRSTStream
      #13: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WriteSettings
      #14: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WriteSettingsAck
      #15: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.Framer.WriteWindowUpdat
e
      #16: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.GoAwayError.Error
      #17: testutil/network/network.go:638:26: network.Network.Cleanup cal
ls log.tmLogger.Error, which eventually calls http2.Setting.String
      #18: testutil/network/network.go:638:26: network.Network.Cleanup cal
ls log.tmLogger.Error, which eventually calls http2.SettingID.String
      #19: testutil/network/util.go:111:45: network.startInProcess calls g
rpc.StartGRPCServer, which eventually calls http2.SettingsFrame.ForeachSe
tting
      #20: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.StreamError.Error
      #21: rpc/websockets.go:319:24: rpc.websocketsServer.tcpGetAndSendRes
ponse calls http.Client.Do, which eventually calls http2.Transport.NewCli
entConn
      #22: rpc/websockets.go:319:24: rpc.websocketsServer.tcpGetAndSendRes
ponse calls http.Client.Do, which eventually calls http2.Transport.RoundT
rip
      #23: rpc/websockets.go:122:31: rpc.websocketsServer.ServeHTTP calls
```

```
websocket.Upgrader.Upgrade, which eventually calls http2.chunkWriter.Writ
e
      #24: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.connError.Error
      #25: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.duplicatePseudoHeaderError.Error
      #26: rpc/websockets.go:324:2: rpc.websocketsServer.tcpGetAndSendResp
onse calls http2.gzipReader.Close
      #27: rpc/websockets.go:326:25: rpc.websocketsServer.tcpGetAndSendRes
ponse calls io.ReadAll, which calls http2.gzipReader.Read
      #28: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.headerFieldNameError.Error
      #29: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.headerFieldValueError.Error
      #30: rpc/websockets.go:319:24: rpc.websocketsServer.tcpGetAndSendRes
ponse calls http.Client.Do, which eventually calls http2.noDialH2RoundTri
pper.RoundTrip
      #31: rpc/backend/blocks.go:588:22: backend.Backend.RPCBlockFromTende
rmintBlock calls http2.pseudoHeaderError.Error
      #32: rpc/websockets.go:122:31: rpc.websocketsServer.ServeHTTP calls
websocket.Upgrader.Upgrade, which eventually calls http2.stickyErrWriter.
Write
      #33: rpc/websockets.go:324:2: rpc.websocketsServer.tcpGetAndSendResp
onse calls http2.transportResponseBody.Close
      #34: rpc/websockets.go:326:25: rpc.websocketsServer.tcpGetAndSendRes
ponse calls io.ReadAll, which calls http2.transportResponseBody.Read
      #35: testutil/network/network.go:638:26: network.Network.Cleanup cal
ls log.tmLogger.Error, which eventually calls http2.writeData.String

Vulnerability #10: GO-2024-2611
    Infinite loop in JSON unmarshaling in google.golang.org/protobuf
  More info: https://pkg.go.dev/vuln/GO-2024-2611
  Module: google.golang.org/protobuf
    Found in: google.golang.org/protobuf@v1.30.0
    Fixed in: google.golang.org/protobuf@v1.33.0
    Example traces found:
      #1: server/json_rpc.go:88:26: server.StartJSONRPC calls http.Server.
Serve, which eventually calls json.Decoder.Peek
      #2: server/json_rpc.go:88:26: server.StartJSONRPC calls http.Server.
Serve, which eventually calls json.Decoder.Read
      #3: testutil/network/network.go:479:31: network.New calls viper.Vipe
r.ReadInConfig, which eventually calls protojson.Unmarshal
      #4: server/json_rpc.go:88:26: server.StartJSONRPC calls http.Server.
Serve, which eventually calls protojson.UnmarshalOptions.Unmarshal

Vulnerability #11: GO-2024-2584
    Slashing evasion in github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-2584
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.10
    Example traces found:
      #1: app/app.go:618:26: app.Tabi.BeginBlocker calls module.Manager.Be
```

```
ginBlock, which eventually calls keeper.Keeper.Slash
      #2: x/captains/types/tx.pb.go:904:19: types.RegisterMsgServer calls
baseapp.MsgServiceRouter.RegisterService, which eventually calls vesting.
msgServer.CreatePeriodicVestingAccount

Vulnerability #12: GO-2024-2572
    Missing BlockedAddressed Validation in Vesting Module in
    github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-2572
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.9
    Example traces found:
      #1: x/captains/types/tx.pb.go:904:19: types.RegisterMsgServer calls
baseapp.MsgServiceRouter.RegisterService, which eventually calls vesting.
msgServer.CreatePeriodicVestingAccount

Vulnerability #13: GO-2024-2571
    Invalid block proposal in github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-2571
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.9
    Example traces found:
      #1: app/app.go:224:28: app.NewTabi calls baseapp.NewBaseApp

Vulnerability #14: GO-2023-2409
    Denial of service when decrypting attacker controlled input in
    github.com/dvsekhvalnov/jose2go
  More info: https://pkg.go.dev/vuln/GO-2023-2409
  Module: github.com/dvsekhvalnov/jose2go
    Found in: github.com/dvsekhvalnov/jose2go@v1.5.0
    Fixed in: github.com/dvsekhvalnov/jose2go@v1.5.1-0.20231206184617-48b
a0b76bc88
    Example traces found:
      #1: rpc/backend/node_info.go:196:47: backend.Backend.ImportRawKey ca
lls keyring.keystore.KeyByAddress, which eventually calls jose2go.Decode
      #2: client/keys/add.go:87:18: keys.RunAddCmd calls keyring.keystore.
Key, which eventually calls jose2go.Encrypt

Vulnerability #15: GO-2023-2153
    Denial of service from HTTP/2 Rapid Reset in google.golang.org/grpc
  More info: https://pkg.go.dev/vuln/GO-2023-2153
  Module: google.golang.org/grpc
    Found in: google.golang.org/grpc@v1.54.0
    Fixed in: google.golang.org/grpc@v1.56.3
    Example traces found:
      #1: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which calls grpc.NewServer
      #2: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls grpc.Server.Serve
      #3: testutil/network/util.go:111:45: network.startInProcess calls gr
pc.StartGRPCServer, which eventually calls transport.NewServerTransport
```

```
Vulnerability #16: GO-2023-1881
    The x/crisis package does not charge ConstantFee in
    github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2023-1881
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: N/A
    Example traces found:
      #1: cmd/tabid/root.go:150:27: tabid.addModuleInitFlags calls crisis.
AddModuleInitFlags
      #2: app/app.go:647:65: app.Tabi.InitChainer calls module.Manager.Get
VersionMap, which calls crisis.AppModule.ConsensusVersion
      #3: app/app.go:623:24: app.Tabi.EndBlocker calls module.Manager.EndB
lock, which calls crisis.AppModule.EndBlock
      #4: app/export.go:44:34: app.Tabi.ExportAppStateAndValidators calls
module.Manager.ExportGenesis, which calls crisis.AppModule.ExportGenesis
      #5: app/app.go:649:27: app.Tabi.InitChainer calls module.Manager.Ini
tGenesis, which calls crisis.AppModule.InitGenesis
      #6: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRout
es, which calls crisis.AppModule.LegacyQuerierHandler
      #7: app/app.go:647:65: app.Tabi.InitChainer calls module.Manager.Get
VersionMap, which calls crisis.AppModule.Name
      #8: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRout
es, which calls crisis.AppModule.QuerierRoute
      #9: app/app.go:527:27: app.NewTabi calls module.Manager.RegisterInva
riants, which calls crisis.AppModule.RegisterInvariants
      #10: app/app.go:530:25: app.NewTabi calls module.Manager.RegisterSer
vices, which calls crisis.AppModule.RegisterServices
      #11: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRou
tes, which calls crisis.AppModule.Route
      #12: testutil/network/network.go:123:53: network.DefaultConfig calls
 module.BasicManager.DefaultGenesis, which calls crisis.AppModuleBasic.De
faultGenesis
      #13: cmd/tabid/root.go:171:35: tabid.queryCommand calls module.Basic
Manager.AddQueryCommands, which calls crisis.AppModuleBasic.GetQueryCmd
      #14: cmd/tabid/root.go:198:32: tabid.txCommand calls module.BasicMan
ager.AddTxCommands, which calls crisis.AppModuleBasic.GetTxCmd
      #15: testutil/network/network.go:123:53: network.DefaultConfig calls
 module.BasicManager.DefaultGenesis, which calls crisis.AppModuleBasic.Na
me
      #16: app/app.go:755:40: app.Tabi.RegisterAPIRoutes calls module.Basi
cManager.RegisterGRPCGatewayRoutes, which calls crisis.AppModuleBasic.Reg
isterGRPCGatewayRoutes
      #17: encoding/config.go:29:23: encoding.MakeConfig calls module.Basi
cManager.RegisterInterfaces, which calls crisis.AppModuleBasic.RegisterIn
terfaces
      #18: encoding/config.go:27:29: encoding.MakeConfig calls module.Basi
cManager.RegisterLegacyAminoCodec, which calls crisis.AppModuleBasic.Regi
sterLegacyAminoCodec
      #19: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which ev
entually calls crisis.AppModuleBasic.ValidateGenesis
      #20: app/modules.go:156:22: app.appModules calls crisis.NewAppModule
```

```
      #21: cmd/tabid/root.go:36:2: tabid.init calls crisis.init

Vulnerability #17: GO-2023-1821
    The x/crisis package does not cause chain halt in
    github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2023-1821
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: N/A
    Example traces found:
      #1: cmd/tabid/root.go:150:27: tabid.addModuleInitFlags calls crisis.
AddModuleInitFlags
      #2: app/app.go:647:65: app.Tabi.InitChainer calls module.Manager.Get
VersionMap, which calls crisis.AppModule.ConsensusVersion
      #3: app/app.go:623:24: app.Tabi.EndBlocker calls module.Manager.EndB
lock, which calls crisis.AppModule.EndBlock
      #4: app/export.go:44:34: app.Tabi.ExportAppStateAndValidators calls
module.Manager.ExportGenesis, which calls crisis.AppModule.ExportGenesis
      #5: app/app.go:649:27: app.Tabi.InitChainer calls module.Manager.Ini
tGenesis, which calls crisis.AppModule.InitGenesis
      #6: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRout
es, which calls crisis.AppModule.LegacyQuerierHandler
      #7: app/app.go:647:65: app.Tabi.InitChainer calls module.Manager.Get
VersionMap, which calls crisis.AppModule.Name
      #8: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRout
es, which calls crisis.AppModule.QuerierRoute
      #9: app/app.go:527:27: app.NewTabi calls module.Manager.RegisterInva
riants, which calls crisis.AppModule.RegisterInvariants
      #10: app/app.go:530:25: app.NewTabi calls module.Manager.RegisterSer
vices, which calls crisis.AppModule.RegisterServices
      #11: app/app.go:528:23: app.NewTabi calls module.Manager.RegisterRou
tes, which calls crisis.AppModule.Route
      #12: testutil/network/network.go:123:53: network.DefaultConfig calls
 module.BasicManager.DefaultGenesis, which calls crisis.AppModuleBasic.De
faultGenesis
      #13: cmd/tabid/root.go:171:35: tabid.queryCommand calls module.Basic
Manager.AddQueryCommands, which calls crisis.AppModuleBasic.GetQueryCmd
      #14: cmd/tabid/root.go:198:32: tabid.txCommand calls module.BasicMan
ager.AddTxCommands, which calls crisis.AppModuleBasic.GetTxCmd
      #15: testutil/network/network.go:123:53: network.DefaultConfig calls
 module.BasicManager.DefaultGenesis, which calls crisis.AppModuleBasic.Na
me
      #16: app/app.go:755:40: app.Tabi.RegisterAPIRoutes calls module.Basi
cManager.RegisterGRPCGatewayRoutes, which calls crisis.AppModuleBasic.Reg
isterGRPCGatewayRoutes
      #17: encoding/config.go:29:23: encoding.MakeConfig calls module.Basi
cManager.RegisterInterfaces, which calls crisis.AppModuleBasic.RegisterIn
terfaces
      #18: encoding/config.go:27:29: encoding.MakeConfig calls module.Basi
cManager.RegisterLegacyAminoCodec, which calls crisis.AppModuleBasic.Regi
sterLegacyAminoCodec
      #19: cmd/tabid/main.go:21:26: tabid.main calls cmd.Execute, which ev
entually calls crisis.AppModuleBasic.ValidateGenesis
```

```
      #20: app/modules.go:156:22: app.appModules calls crisis.NewAppModule
      #21: cmd/tabid/root.go:36:2: tabid.init calls crisis.init

Your code is affected by 17 vulnerabilities from 8 modules and the Go stan
dard library.
This scan also found 2 vulnerabilities in packages you import and 11
vulnerabilities in modules you require, but your code doesn't appear to ca
ll
these vulnerabilities.
```

The following dependencies are recommended for priority attention:

```
Vulnerability #4: GO-2024-3279
    Mismatched bit-length validation in can lead to panic in cosmossdk.io/
math
  More info: https://pkg.go.dev/vuln/GO-2024-3279
  Module: cosmossdk.io/math
    Found in: cosmossdk.io/math@v1.0.0-rc.0
    Fixed in: cosmossdk.io/math@v1.4.0
    Example traces found:
      #1: x/feemarket/types/params.go:6:2: types.init calls math.init, whi
ch eventually calls math.LegacyDec.Quo
```

```
Vulnerability #8: GO-2024-2694
    Potential Reentrancy using Timeout Callbacks in ibc-hooks in
    github.com/cosmos/ibc-go
  More info: https://pkg.go.dev/vuln/GO-2024-2694
  Module: github.com/cosmos/ibc-go/v6
    Found in: github.com/cosmos/ibc-go/v6@v6.1.1
    Fixed in: github.com/cosmos/ibc-go/v6@v6.3.0
    Example traces found:
      #1: app/ante/evm/eth.go:377:13: evm.EthIncrementSenderSequenceDecora
tor.AnteHandle calls types.ChainAnteDecorators, which eventually calls ke
eper.Keeper.Timeout
      #2: app/ante/evm/eth.go:377:13: evm.EthIncrementSenderSequenceDecora
tor.AnteHandle calls types.ChainAnteDecorators, which eventually calls ke
eper.Keeper.TimeoutOnClose
```

```
Vulnerability #11: GO-2024-2584
    Slashing evasion in github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-2584
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.10
    Example traces found:
      #1: app/app.go:618:26: app.Tabi.BeginBlocker calls module.Manager.Be
ginBlock, which eventually calls keeper.Keeper.Slash
      #2: x/captains/types/tx.pb.go:904:19: types.RegisterMsgServer calls
baseapp.MsgServiceRouter.RegisterService, which eventually calls vesting.
msgServer.CreatePeriodicVestingAccount
```

```
Vulnerability #12: GO-2024-2572
    Missing BlockedAddressed Validation in Vesting Module in
    github.com/cosmos/cosmos-sdk
  More info: https://pkg.go.dev/vuln/GO-2024-2572
  Module: github.com/cosmos/cosmos-sdk
    Found in: github.com/cosmos/cosmos-sdk@v0.46.15
    Fixed in: github.com/cosmos/cosmos-sdk@v0.47.9
    Example traces found:
      #1: x/captains/types/tx.pb.go:904:19: types.RegisterMsgServer calls
baseapp.MsgServiceRouter.RegisterService, which eventually calls vesting.
msgServer.CreatePeriodicVestingAccount
```

Result: Confirmed

Fix Result: Acknowledged

4. CONCLUSION

In this audit, we thoroughly analyzed **tabi** Blockchain implementation. The problems found are described and explained in detail in Section 3. The problems found in the audit have been communicated to the project leader. We therefore consider the audit result to be **PASSED**. To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, audit findings, or potential gaps in scope/coverage.

5. APPENDIX

5.1 Basic Coding Assessment

5.1.1 Apply Verification Control

- Description: The security of apply verification
- Result: Not found
- Severity: Critical

5.1.2 Authorization Access Control

- Description: Permission checks for external integral functions
- Result: Not found
- Severity: Critical

5.1.3 Forged Transfer Vulnerability

- Description: Assess whether there is a forged transfer notification vulnerability in the contract
- Result: Not found
- Severity: Critical

5.1.4 Transaction Rollback Attack

- Description: Assess whether there is transaction rollback attack vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.5 Transaction Block Stuffing Attack

- Description: Assess whether there is transaction blocking attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.6 Soft Fail Attack Assessment

- Description: Assess whether there is soft fail attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.7 Hard Fail Attack Assessment

- Description: Examine for hard fail attack vulnerability
- Result: Not found
- Severity: Critical

5.1.8 Abnormal Memo Assessment

- Description: Assess whether there is abnormal memo vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.9 Abnormal Resource Consumption

- Description: Examine whether abnormal resource consumption in contract processing.
- Result: Not found
- Severity: Critical

5.1.10 Random Number Security

- Description: Examine whether the code uses insecure random number.
- Result: Not found
- Severity: Critical

5.2 Advanced Code Scrutiny

5.2.1 Cryptography Security

- Description: Examine for weakness in cryptograph implementation.
- Results: Not Found
- Severity: High

5.2.2 Account Permission Control

- Description: Examine permission control issue in the contract
- Results: Not Found
- Severity: Medium

5.2.3 Malicious Code Behavior

- Description: Examine whether sensitive behavior present in the code
- Results: Not found
- Severity: Medium

5.2.4 Sensitive Information Disclosure

- Description: Examine whether sensitive information disclosure issue present in the code.
- Result: Not found
- Severity: Medium

5.2.5 System API

- Description: Examine whether system API application issue present in the code
- Results: Not found
- Severity: Low

6. DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without ExVul's prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts ExVul to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. ExVul's position is that each company and individual are responsible for their own due diligence and continuous security. ExVul's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

7. REFERENCES

[1] MITRE. CWE- 191: Integer Underflow (Wrap or Wraparound).

https://cwe.mitre.org/data/ definitions/191.html.

[2] MITRE. CWE- 197: Numeric Truncation Error.

https://cwe.mitre.org/data/definitions/197. html.

[3] MITRE. CWE-400: Uncontrolled Resource Consumption.

https://cwe.mitre.org/data/ definitions/400.html.

[4] MITRE. CWE-440: Expected Behavior Violation.

https://cwe.mitre.org/data/definitions/440. html.

[5] MITRE. CWE-684: Protection Mechanism Failure.

https://cwe.mitre.org/data/definitions/ 693.html.

[6] MITRE. CWE CATEGORY: 7PK - Security Features.

https://cwe.mitre.org/data/definitions/ 254.html.

[7] MITRE. CWE CATEGORY: Behavioral Problems.

https://cwe.mitre.org/data/definitions/438. html.

[8] MITRE. CWE CATEGORY: Numeric Errors.

https://cwe.mitre.org/data/definitions/189.html.

[9] MITRE. CWE CATEGORY: Resource Management Errors.

https://cwe.mitre.org/data/ definitions/399.html.

[10] OWASP. Risk Rating Methodology.

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

www.exvul.com

contact@exvul.com

@EXVULSEC

github.com/EXVUL-Sec

ExVul