// HALBORN

Entangle - Blockchain

Cosmos Security Assessment

Prepared by: **Halborn**

Date of Engagement: **December 21st, 2023 - January 15th, 2024**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE
0.1	Document Creation	01/12/2024
0.2	Document Updates	01/14/2024
0.3	Draft Review	01/16/2024
1.0	Remediation Plan	02/08/2024
1.1	Remediation Plan Review	02/09/2024

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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Entangle engaged Halborn to conduct a security assessment on their modules, beginning on December 21st, 2023 and ending on January 15th, 2024. The security assessment was scoped to the sections of code that pertain to the **Cosmos Appchain**.

1.2 ASSESSMENT SUMMARY

The team at Halborn was provided one month for the engagement and assigned one full-time security engineer to verify the security of the merge requests. 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 assessment is to:

- Ensure that the **Entangle Modules** operate as intended.
- Identify potential security issues with the custom modules used in the Cosmos AppChain.

In summary, Halborn identified some improvements to reduce the likelihood and impact of risks that were mostly addressed by the Entangle 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 assessment:

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

2. RISK METHODOLOGY

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

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

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

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

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

2.1 EXPLOITABILITY

Attack Origin (AO):

Captures whether the attack requires compromising a specific account.

Attack Cost (AC):

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

Attack Complexity (AX):

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

Metrics:

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

Exploitability E is calculated using the following formula:

$$E = \prod m_e$$

2.2 IMPACT

Confidentiality (C):

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

Integrity (I):

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

Availability (A):

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

Deposit (D):

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

Yield (Y):

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

Metrics:

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

Impact I is calculated using the following formula:

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

2.3 SEVERITY COEFFICIENT

Reversibility (R):

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

Scope (S):

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

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

Severity Coefficient C is obtained by the following product:

$$C = rs$$

The Vulnerability Severity Score S is obtained by:

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

The score is rounded up to 1 decimal places.

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

2.4 SCOPE

This review was scoped to the entangle-blockchain repository.

1. IN-SCOPE TREE & COMMIT :

- entangle-blockchain

Commit ID : cf6116a40ada252b2561d4e9f9d8023b4378e4fe

REMEDIATION COMMIT IDs :

- 2f9e924f95040782816d9d915d9ada0861e075ae
- dda8d04e01c0e03f2eaa221b5242379640a1e758
- 026a2a82718060c8b1034a3bf675d5fbe039f4bf
- 5312ceac9a843e782d5d892e0d52e8dd8b1f8368

EXECUTIVE OVERVIEW

3. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
1	4	1	2	3

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) MISSING CODEC HANDLER FOR DISTRIBUTOR MESSAGES	Critical (10)	SOLVED - 01/12/2024
(HAL-02) MISSING VALIDATION FOR END TIME IN ADDDISTRIBUTOR FUNCTION LEADS TO EXPIRED DISTRIBUTOR	High (7.5)	SOLVED - 01/12/2024
(HAL-03) UNHANDLED RETURN IN REMOVEADMIN FUNCTION IMPLEMENTATION	High (8.1)	SOLVED - 02/08/2024
(HAL-04) ERROR HANDLING IN ADDADMIN METHOD OF KEEPER MODULE	High (8.1)	SOLVED - 02/08/2024
(HAL-05) ERROR HANDLING IN REMOVEDISTRIBUTOR METHOD OF MSGSERVER	High (8.1)	SOLVED - 02/08/2024
(HAL-06) END DATE IS NOT CHECKED ON THE PROPOSALS	Medium (5.0)	SOLVED - 01/12/2024
(HAL-07) DOCKER IMAGE RUNNING AS ROOT	Low (3.8)	SOLVED - 01/12/2024
(HAL-08) LACK OF SENTRY NODE INFRASTRUCTURE	Low (3.8)	SOLVED - 02/08/2024
(HAL-09) LACK OF SPEC ON THE MODULE	Informational (0.0)	ACKNOWLEDGED
(HAL-10) OUT OF DATE GO-ETHEREUM	Informational (0.0)	ACKNOWLEDGED
(HAL-11) LACK OF DEBUG TRACE CALL SUPPORT	Informational (0.0)	ACKNOWLEDGED

FINDINGS & TECH DETAILS

4.1 (HAL-01) MISSING CODEC HANDLER FOR DISTRIBUTOR MESSAGES - CRITICAL(10)

Description:

The implementation lacks a codec handler registration for the all distributor message in the amino codec. This will lead to issues when encoding or decoding messages related to the distributor messages. To address this issue, the distributor messages should be registered in the RegisterInterfaces function. This ensures that the amino codec has the necessary handler for encoding and decoding this message type.

Code Location:

/x/distributorsauth/types/codec.go

Listing 1

```go
package types

import (
    "github.com/cosmos/cosmos-sdk/codec"
    cdctypes "github.com/cosmos/cosmos-sdk/codec/types"
    govtypes "github.com/cosmos/cosmos-sdk/x/gov/types/v1beta1"

    // this line is used by starport scaffolding # 1
    "github.com/cosmos/cosmos-sdk/types/msgservice"
)

func RegisterCodec(cdc *codec.LegacyAmino) {
    // this line is used by starport scaffolding # 2
}

func RegisterInterfaces(registry cdctypes.InterfaceRegistry) {
    // this line is used by starport scaffolding # 3

    msgservice.RegisterMsgServiceDesc(registry, &_Msg_serviceDesc)
    registry.RegisterImplementations((*govtypes.Content)(nil), &AddDistributorProposal{})

}

var (
    Amino     = codec.NewLegacyAmino()
    ModuleCdc = codec.NewProtoCodec(cdctypes.NewInterfaceRegistry())
)
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:C/D:N/Y:N/R:N/S:U (10)

Recommendation:

To address this issue, the distributor messages should be registered in the RegisterInterfaces function. This ensures that the amino codec has the necessary handler for encoding and decoding this message type.

Remediation Plan:

SOLVED: The Entangle team solved the issue by registering the codecs.

Commit ID: 2f9e924f95040782816d9d915d9ada0861e075ae

4.2 (HAL-02) MISSING VALIDATION FOR END TIME IN ADDDISTRIBUTOR FUNCTION LEADS TO EXPIRED DISTRIBUTOR - HIGH (7.5)

Description:

The AddDistributor function in the current implementation lacks validation for the EndDate parameter of a distributor. This omission can lead to scenarios where a distributor is added with an already expired EndDate, potentially causing issues in the management and operation of distributors. Without proper validation, the system might incorrectly process or recognize these distributors, leading to inconsistencies or errors in the distribution process. - Introduce a check to verify that the EndDate for a new distributor is in the future relative to the current block time.

This can be achieved by comparing msg.EndDate with ctx.BlockTime() within the AddDistributor function.

Code Location:

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

Listing 2

```
func (s msgServer) AddDistributor(goCtx context.Context, msg *types.MsgAddDistributor) (*types.MsgAddDistributorResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    err := s.checkSenderHaveAdminsWrights(ctx, msg.Sender, false)
    if err != nil {
        return nil, err
    }

    var DistributorInfo = types.DistributorInfo{
        Address: msg.DistributorAddress,
        EndDate: msg.EndDate,
    }

    ctx.BlockTime()

    s.Keeper.AddDistributor(ctx, DistributorInfo)

    return &types.MsgAddDistributorResponse{}, nil
}
```

Proof Of Concept:

Listing 3

```
func (suite *KeeperTestSuite) TestAddDistributor() {

    testCases := []struct {
        name                string
        malleate            func(string)
        sender              string
        distributor_address string
        end_date            uint64
        success             bool
    }{
        {
            "Add distributor success by Admin",
            func(addr string) {
                suite.app.DistributorsAuthKeeper.AddAdmin(suite.ctx, types.Admin{Address: addr, EditOption: false})
            },
            "ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
            "ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
            uint64(1234),
            true,
        },
        {
            "Add distributor failed by Distributor",
            func(addr string) {
                suite.app.DistributorsAuthKeeper.AddDistributor(suite.ctx, types.DistributorInfo{Address: addr, EndDate: uint64(0)})
            },
            "ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
            "ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
            uint64(1234),
            false,
        },
        {
            "Add distributor failed by guest",
            func(addr string) {},
            "ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
            "ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
            uint64(1234),
            false,
        },
    }

    for _, tc := range testCases {
        suite.Run(tc.name, func() {
            suite.SetupTest()

            tc.malleate(tc.sender)
            _, add_err := suite.msgServer.AddDistributor(suite.ctx
↳ , &types.MsgAddDistributor{Sender: tc.sender, DistributorAddress:
↳ tc.distributor_address, EndDate: tc.end_date})

            distr, err := suite.app.DistributorsAuthKeeper.
↳ GetDistributor(suite.ctx, tc.distributor_address)
            if !tc.success {
                suite.Require().Error(err)
                return
            }
            suite.Require().NoError(err)
            suite.Require().NoError(add_err)
            suite.Require().Equal(distr, types.DistributorInfo{
↳ Address: tc.distributor_address, EndDate: tc.end_date})
        })
    }
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:H/A:N/D:N/Y:N/R:N/S:U (7.5)

Recommendation:

To address this issue, consider adding validation on the end date.

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding validation.

Commit ID: dda8d04e01c0e03f2eaa221b5242379640a1e758

4.3 (HAL-03) UNHANDLED RETURN IN REMOVEADMIN FUNCTION IMPLEMENTATION - HIGH (8.1)

Description:

The RemoveAdmin function is currently set up to remove an admin, but it does not handle a specific error scenario adequately. This oversight could lead to silent failures in certain cases. In the event of an error in address conversion, the function exits silently, leading to a lack of clarity about whether the operation succeeded or failed.

Code Location:

/x/distributorsauth/keeper/msg_server.go#L74

Listing 4

```go
func (s msgServer) RemoveAdmin(goCtx context.Context, msg *types.MsgRemoveAdmin) (*types.MsgRemoveAdminResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    err := s.checkSenderHaveAdminsWrights(ctx, msg.Sender, true)
    if err != nil {
        return nil, err
    }

    s.Keeper.RemoveAdmin(ctx, msg.AdminAddress)

    return &types.MsgRemoveAdminResponse{}, nil
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:H/A:L/D:N/Y:N/R:N/S:U (8.1)

Recommendation:

Update the RemoveAdmin function to effectively handle any relay errors.

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding return value.

Commit ID: 5312ceac9a843e782d5d892e0d52e8dd8b1f8368

4.4 (HAL-04) ERROR HANDLING IN ADDADMIN METHOD OF KEEPER MODULE - HIGH (8.1)

Description:

A potential oversight in the error handling mechanism has been observed in the AddAdmin method within our Keeper module. While the Keeper's AddAdmin function correctly handles errors, its caller method in the msgServer struct does not properly handle the error that might be returned.

The error returned by Keeper.AddAdmin is not handled in msgServer.AddAdmin. This could lead to scenarios where the function fails silently, and the admin is not added, but the caller is unaware of the failure.

Code Location:

/x/distributorsauth/keeper/msg_server.go#L69

Listing 5

```
func (s msgServer) AddAdmin(goCtx context.Context, msg *types.MsgAddAdmin) (*types.MsgAddAdminResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    err := s.checkSenderHaveAdminsWrights(ctx, msg.Sender, true)
    if err != nil {
        return nil, err
    }

    var AdminInfo = types.Admin{
        Address:    msg.AdminAddress,
        EditOption: msg.EditOption,
    }

    s.Keeper.AddAdmin(ctx, AdminInfo)

    return &types.MsgAddAdminResponse{}, nil
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:H/A:L/D:N/Y:N/R:N/S:U (8.1)

Recommendation:

Modify the msgServer.AddAdmin method to handle and propagate the error returned by Keeper.AddAdmin. This can be achieved by checking the error and returning it if not nil.

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding return value.

Commit ID: 5312ceac9a843e782d5d892e0d52e8dd8b1f8368

4.5 (HAL-05) ERROR HANDLING IN REMOVEDISTRIBUTOR METHOD OF MSGSERVER - HIGH (8.1)

Description:

An issue has been identified in the RemoveDistributor method within the msgServer struct. The method is designed to remove a distributor from the system. However, there is a lack of error handling for the response from the RemoveDistributor function of the Keeper.

The primary concern is the absence of error handling for the Keeper.RemoveDistributor call. If an error occurs during the removal process, it is not captured or relayed back to the caller. This can lead to silent failures, where the caller is not informed if the distributor was not successfully removed.

Code Location:

/x/distributorsauth/keeper/msg_server.go#L51

Listing 6

```go
func (s msgServer) RemoveDistributor(goCtx context.Context, msg *types.MsgRemoveDistributor) (*types.MsgRemoveDistributorResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    err := s.checkSenderHaveAdminsWrights(ctx, msg.Sender, false)
    if err != nil {
        return nil, err
    }

    s.Keeper.RemoveDistributor(ctx, msg.DistributorAddress)

    return &types.MsgRemoveDistributorResponse{}, nil
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:H/A:L/D:N/Y:N/R:N/S:U (8.1)

Recommendation:

Amend the msgServer.RemoveDistributor method to capture and return any errors that occur during the call to Keeper.RemoveDistributor. This involves checking for an error after the call and, if present, returning it to the caller.

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding return value.

Commit ID: 5312ceac9a843e782d5d892e0d52e8dd8b1f8368

4.6 (HAL-06) END DATE IS NOT CHECKED ON THE PROPOSALS - MEDIUM (5.0)

Description:

The AddDistributor function in our existing system currently lacks critical validation for the EndDate parameter of a distributor. This gap in the validation process can lead to a situation where distributors are added with an EndDate that has already passed.

Code Location:

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

Listing 7

```go
func (s msgServer) AddDistributor(goCtx context.Context, msg *types.MsgAddDistributor) (*types.MsgAddDistributorResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    err := s.checkSenderHaveAdminsWrights(ctx, msg.Sender, false)
    if err != nil {
        return nil, err
    }

    var DistributorInfo = types.DistributorInfo{
        Address: msg.DistributorAddress,
        EndDate: msg.EndDate,
    }

    ctx.BlockTime()

    s.Keeper.AddDistributor(ctx, DistributorInfo)

    return &types.MsgAddDistributorResponse{}, nil
}
```

Proof Of Concept:

Listing 8

```go
func (suite *KeeperTestSuite) TestAddDistributor() {

	testCases := []struct {
		name                string
		malleate            func(string)
		sender              string
		distributor_address string
		end_date            uint64
		success             bool
	}{
		{
			"Add distributor success by Admin",
			func(addr string) {
				suite.app.DistributorsAuthKeeper.AddAdmin(suite.ctx, types.Admin{Address: addr, EditOption: false})
			},
			"ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
			"ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
			uint64(1234),
			true,
		},
		{
			"Add distributor failed by Distributor",
			func(addr string) {
				suite.app.DistributorsAuthKeeper.AddDistributor(suite.ctx, types.DistributorInfo{Address: addr, EndDate: uint64(0)})
			},
			"ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
			"ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
			uint64(1234),
			false,
		},
		{
			"Add distributor failed by guest",
			func(addr string) {},
			"ethm1cdsdkvxydypnhtec5y880qdtdexcu2ehf0lpv8",
			"ethm1trhgn3un9wqlxhxwxspxaaalnynr4539v8vdmc",
			uint64(1234),
			false,
		},
    }

    for _, tc := range testCases {
        suite.Run(tc.name, func() {
            suite.SetupTest()

            tc.malleate(tc.sender)
            _, add_err := suite.msgServer.AddDistributor(suite.ctx, &types.MsgAddDistributor{Sender: tc.sender, DistributorAddress: tc.distributor_address, EndDate: tc.end_date})

            distr, err := suite.app.DistributorsAuthKeeper.GetDistributor(suite.ctx, tc.distributor_address)
            if !tc.success {
                suite.Require().Error(err)
                return
            }
            suite.Require().NoError(err)
            suite.Require().NoError(add_err)
            suite.Require().Equal(distr, types.DistributorInfo{Address: tc.distributor_address, EndDate: tc.end_date})
        })
    }
}
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:M/A:N/D:N/Y:N/R:N/S:U (5.0)

Recommendation:

To address this issue, consider adding validation on the end date for the proposals.

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding validation.

Commit ID: dda8d04e01c0e03f2eaa221b5242379640a1e758

4.7 (HAL-07) DOCKER IMAGE RUNNING AS ROOT - LOW (3.8)

Description:

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

Code Location:

Dockerfile

Listing 9: Dockerfile

```
FROM golang:alpine AS build-env

# Set up dependencies
ENV PACKAGES git build-base

# Set working directory for the build
WORKDIR /node

# Install dependencies
RUN apk add --update $PACKAGES
RUN apk add linux-headers

RUN apk add go
RUN apk add make

# ARG key_password


# Add source files
COPY . .

# Make the binary
RUN make build
```

```

# Final image
FROM alpine:3.18.5

# Install ca-certificates
RUN apk add --update ca-certificates jq
WORKDIR /node

# Copy over binaries from the build-env
COPY --from=build-env /node/build/entangled /usr/bin/entangled

WORKDIR /

COPY . .

RUN chmod +x run_node.sh

ENTRYPOINT ["/run_node.sh"]
```

BVSS:

AO:A/AC:L/AX:M/C:M/I:L/A:N/D:N/Y:N/R:N/S:U (3.8)

Recommendation:

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

Listing 10: Reference

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

Remediation Plan:

SOLVED: The Entangle team solved the issue by adding non-root user.

Commit ID: 026a2a82718060c8b1034a3bf675d5fbe039f4bf

4.8 (HAL-08) LACK OF SENTRY NODE INFRASTRUCTURE - LOW (3.8)

Description:

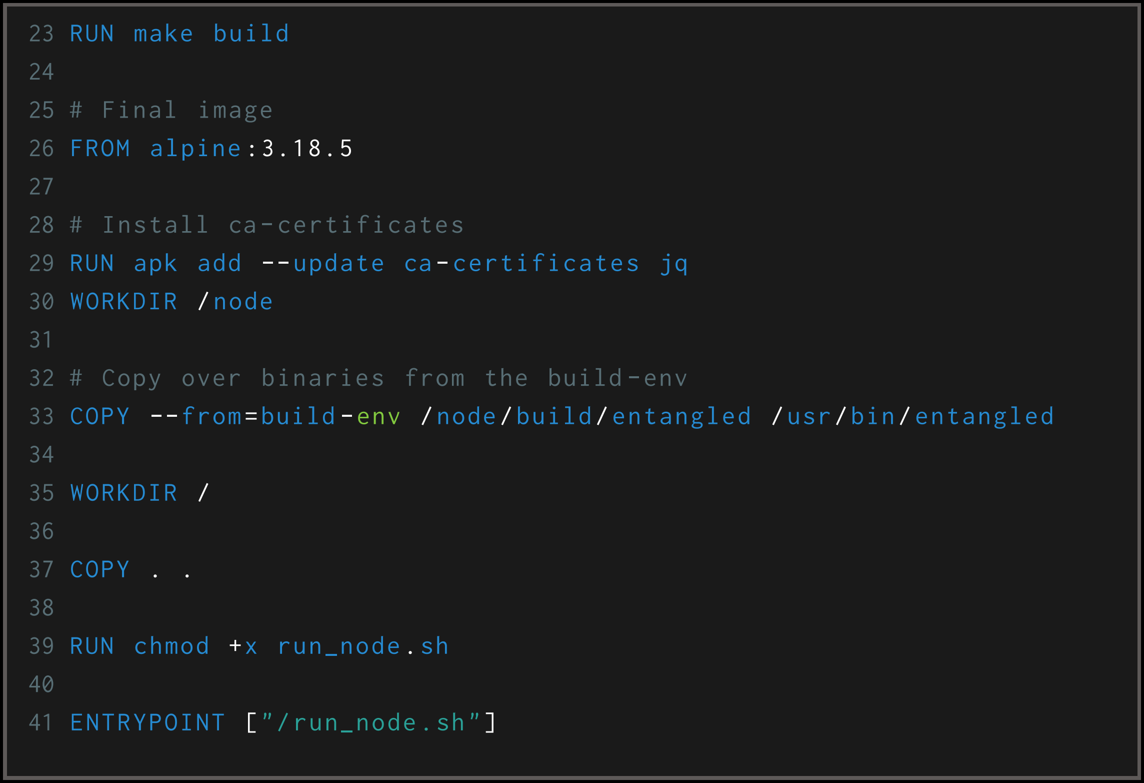
The Sentry Node Architecture is an infrastructure example for DDoS mitigation on validator nodes. To mitigate the issue, multiple distributed nodes (sentry nodes) are deployed in cloud environments. With the possibility of easy scaling, it is harder to make an impact on the validator node. New sentry nodes can be brought up during a DDoS attack, and using the gossip network they can be integrated into the transaction flow.

Code Location:

Dockerfile

Listing 11: Dockerfile

```
FROM golang:alpine AS build-env

# Set up dependencies
ENV PACKAGES git build-base

# Set working directory for the build
WORKDIR /node

# Install dependencies
RUN apk add --update $PACKAGES
RUN apk add linux-headers

RUN apk add go
RUN apk add make

# ARG key_password


# Add source files
COPY . .

# Make the binary
```



```
RUN make build

# Final image
FROM alpine:3.18.5

# Install ca-certificates
RUN apk add --update ca-certificates jq
WORKDIR /node

# Copy over binaries from the build-env
COPY --from=build-env /node/build/entangled /usr/bin/entangled

WORKDIR /

COPY . .

RUN chmod +x run_node.sh

ENTRYPOINT ["/run_node.sh"]
```

BVSS:

AO:A/AC:L/AX:M/C:M/I:L/A:N/D:N/Y:N/R:N/S:U (3.8)

Recommendation:

Consider adding sentry node infrastructure for app-chain.

Remediation Plan:

SOLVED: The Entangle team solved the issue by designing sentry node infrastructure.

4.9 (HAL-09) LACK OF SPEC ON THE MODULE - INFORMATIONAL (0.0)

Description:

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

Code Location:

distributorsauth

BVSS:

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

Recommendation:

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

Remediation Plan:

ACKNOWLEDGED: The Entangle team acknowledged this finding.

4.10 (HAL-10) OUT OF DATE GO-ETHEREUM - INFORMATIONAL (0.0)

Description:

During the code review, It has been noticed that Go-ethereum version is not updated with the recent versions.

Example update can be seen from the following link.

Code Location:

go.mod

Listing 12: Dockerfile

```
FROM golang:alpine AS build-env

# Set up dependencies
ENV PACKAGES git build-base

# Set working directory for the build
WORKDIR /node

# Install dependencies
RUN apk add --update $PACKAGES
RUN apk add linux-headers

RUN apk add go
RUN apk add make

# ARG key_password


# Add source files
COPY . .

# Make the binary
RUN make build

```

```
# Final image
FROM alpine:3.18.5

# Install ca-certificates
RUN apk add --update ca-certificates jq
WORKDIR /node

# Copy over binaries from the build-env
COPY --from=build-env /node/build/entangled /usr/bin/entangled

WORKDIR /

COPY . .

RUN chmod +x run_node.sh

ENTRYPOINT ["/run_node.sh"]
```

BVSS:

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

Recommendation:

It is recommended to update go-ethereum version.

Remediation Plan:

ACKNOWLEDGED: The Entangle team acknowledged this finding.

4.11 (HAL-11) LACK OF DEBUG TRACE CALL SUPPORT - INFORMATIONAL (0.0)

Description:

The debug_traceCall method is commonly used for transaction simulation and plays a critical role in providing detailed insights into transaction execution, especially for wallet applications. It allows users to preview the outcome of a transaction before actual execution, enhancing transparency and predictability.

BVSS:

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

Recommendation:

Consider implementing a debug trace call. A sample implementation can be seen from below :

- Debug trace call support

Remediation Plan:

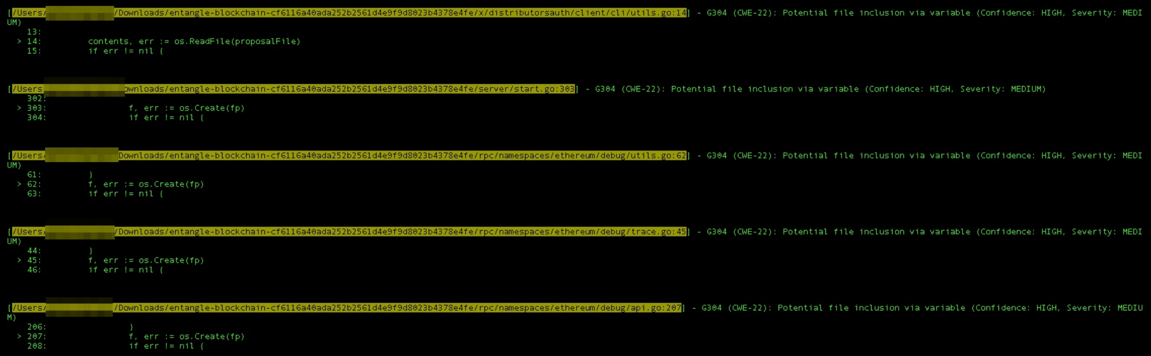
ACKNOWLEDGED: The Entangle team acknowledged this finding.

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.

Gosec - Analysis Output Sample:



```
[/Users/[redacted]/Downloads/entangle-blockchain-cf6116a40ada252b2561d4e9f9d8023b4378e4fe/x/distributorsauth/client/cli/utils.go:14] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDI
UM)
    13:
  > 14:         contents, err := os.ReadFile(proposalFile)
    15:         if err != nil {


[/Users[redacted]ownloads/entangle-blockchain-cf6116a40ada252b2561d4e9f9d8023b4378e4fe/server/start.go:303] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
    302:
  > 303:                f, err := os.Create(fp)
    304:                if err != nil {


[/Users/[redacted]Downloads/entangle-blockchain-cf6116a40ada252b2561d4e9f9d8023b4378e4fe/rpc/namespaces/ethereum/debug/utils.go:62] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDI
UM)
    61:         }
  > 62:         f, err := os.Create(fp)
    63:         if err != nil {


[/Users/[redacted]/Downloads/entangle-blockchain-cf6116a40ada252b2561d4e9f9d8023b4378e4fe/rpc/namespaces/ethereum/debug/trace.go:45] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDI
UM)
    44:         }
  > 45:         f, err := os.Create(fp)
    46:         if err != nil {


[/Users/[redacted]/Downloads/entangle-blockchain-cf6116a40ada252b2561d4e9f9d8023b4378e4fe/rpc/namespaces/ethereum/debug/api.go:207] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIU
M)
    206:                }
  > 207:                f, err := os.Create(fp)
    208:                if err != nil {
```

Semgrep - Security Analysis Output Sample:

Command :

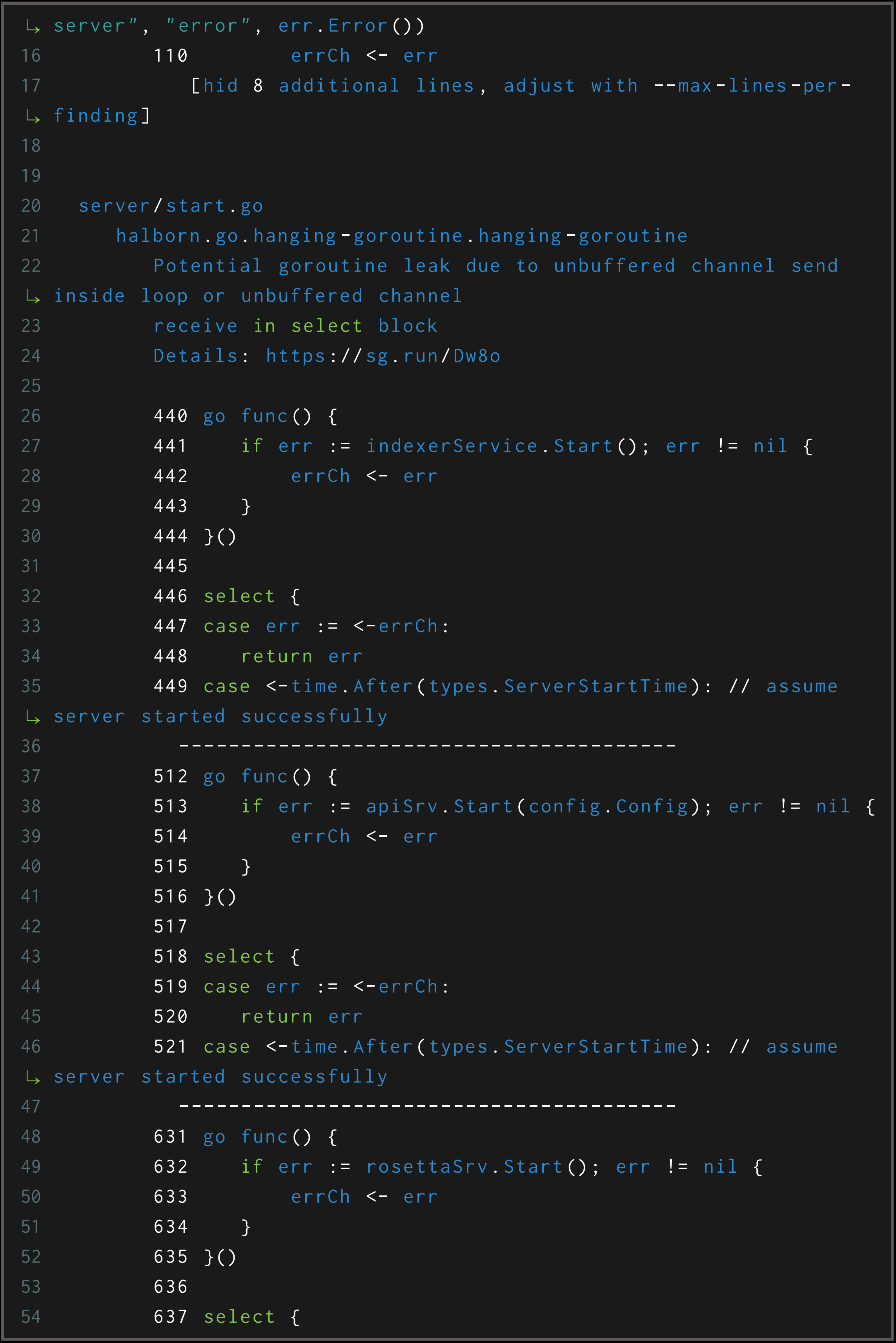
Listing 13

```
semgrep --config configFile
```

Output :

Listing 14

```
  server/json_rpc.go
     halborn.go.hanging-goroutine.hanging-goroutine
        Potential goroutine leak due to unbuffered channel send inside loop or unbuffered channel
        receive in select block
        Details: https://sg.run/Dw8o

        101 go func() {
        102     ctx.Logger.Info("Starting JSON-RPC server", "address", config.JSONRPC.Address)
        103     if err := httpSrv.Serve(ln); err != nil {
        104         if err == http.ErrServerClosed {
        105             close(httpSrvDone)
        106             return
        107         }
        108
        109         ctx.Logger.Error("failed to start JSON-RPC
```



```
↳ server", "error", err.Error())
        110         errCh <- err
           [hid 8 additional lines, adjust with --max-lines-per-
↳ finding]


  server/start.go
     halborn.go.hanging-goroutine.hanging-goroutine
        Potential goroutine leak due to unbuffered channel send
↳ inside loop or unbuffered channel
        receive in select block
        Details: https://sg.run/Dw8o

        440 go func() {
        441    if err := indexerService.Start(); err != nil {
        442        errCh <- err
        443    }
        444 }()
        445
        446 select {
        447 case err := <-errCh:
        448    return err
        449 case <-time.After(types.ServerStartTime): // assume
↳ server started successfully
          ----------------------------------------
        512 go func() {
        513    if err := apiSrv.Start(config.Config); err != nil {
        514        errCh <- err
        515    }
        516 }()
        517
        518 select {
        519 case err := <-errCh:
        520    return err
        521 case <-time.After(types.ServerStartTime): // assume
↳ server started successfully
          ----------------------------------------
        631 go func() {
        632    if err := rosettaSrv.Start(); err != nil {
        633        errCh <- err
        634    }
        635 }()
        636
        637 select {
```

```
        638 case err := <-errCh:
        639     return err
        640 case <-time.After(types.ServerStartTime): // assume server started successfully


  testutil/network/network.go
     halborn.go.missing-unlock-before-return.missing-unlock-before-return
        Missing mutex unlock before returning from a function. This could result in panics
        resulting from double lock operations
        Details: https://sg.run/18Bk

        238 return nil, fmt.Errorf("invalid chain-id: %s", cfg.ChainID)
          ----------------------------------------
        290 return nil, err
          ----------------------------------------
        297 return nil, err
          ----------------------------------------
        306 return nil, err
          ----------------------------------------
        316 return nil, err
          ----------------------------------------
        324 return nil, err
          ----------------------------------------
        334 return nil, err
          ----------------------------------------
        357 return nil, err
          ----------------------------------------
        362 return nil, err
          ----------------------------------------
        371 return nil, err
          ----------------------------------------
        377 return nil, err
          ----------------------------------------
        385 return nil, err
          ----------------------------------------
        392 return nil, err
          ----------------------------------------
        398 return nil, err
          ----------------------------------------
        403 return nil, err
```

```
          ----------------------------------------
        415 return nil, err
          ----------------------------------------
        421 return nil, err
          ----------------------------------------
        438 return nil, err
          ----------------------------------------
        450 return nil, err
          ----------------------------------------
        455 return nil, err
          ----------------------------------------
        463 return nil, err
          ----------------------------------------
        477 return nil, err
          ----------------------------------------
        482 return nil, err
          ----------------------------------------
        486 return nil, err
          ----------------------------------------
        497 return nil, err
          ----------------------------------------
        529 return nil, err
          ----------------------------------------
        533 return nil, err
          ----------------------------------------
        540 return nil, err
          ----------------------------------------
        550 return network, nil


  testutil/network/util.go
     halborn.go.hanging-goroutine.hanging-goroutine
        Potential goroutine leak due to unbuffered channel send
↳ inside loop or unbuffered channel
        receive in select block
        Details: https://sg.run/Dw8o

        110 go func() {
        111    if err := apiSrv.Start(val.AppConfig.Config); err
↳ != nil {
        112        errCh <- err
        113    }
        114 }()
        115
        116 select {
        117 case err := <-errCh:
        118     return err
        119 case <-time.After(srvtypes.ServerStartTime): // assume server started successfully


  x/evm/statedb/statedb.go
     halborn.go.invalid-usage-of-modified-variable.invalid-usage-of-modified-variable
        Variable `newObj` is likely modified and later used on error. In some cases this could
        result  in panics due to a nil dereference
        Details: https://sg.run/WWQ2

        273 newObj, prev := s.createObject(addr)
        274 if prev != nil {
        275     newObj.setBalance(prev.account.Balance)
        276 }
```

THANK YOU FOR CHOOSING

HALBORN