Archway - Tracking and Rewards

Cosmos Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 16th, 2022 - October 28th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	10/25/2022	Chris Meistre
0.2	Document Updates	10/26/2022	Gokberk Gulgun
0.3	Draft Review	10/27/2022	Gabi Urrutia
1.0	Remediation Plan	02/21/2022	Gokberk Gulgun
1.1	Remediation Plan Review	02/22/2022	Gabi Urrutia

CONTACTS

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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Archway engaged Halborn to conduct a security audit on their tracking and rewards modules, beginning on October 16th, 2022 and ending on October 28th, 2022 . The security assessment was scoped to the GitHub repository provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn assigned one full-time security engineer to audit the security of the tracking and rewards modules. 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 to achieve the following:

- Ensure that the Tracking and Rewards modules function as intended.
- Identify potential security issues with the Archway Team.

In summary, Halborn identified some security risks that were accepted and acknowledged by the Archway 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 tracking and rewards 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. (staticcheck, gosec, unconvert, LGTM, ineffassign and semgrep)
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.
- Dynamic Analysis on the Tracking and Rewards modules functions and data types.

RISK METHODOLOGY:

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

RISK SCALE - LIKELIHOOD

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

RISK SCALE - IMPACT

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

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

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

The security assessment was scoped to the following respositories:
- archway-network/archway

IN-SCOPE MODULES:

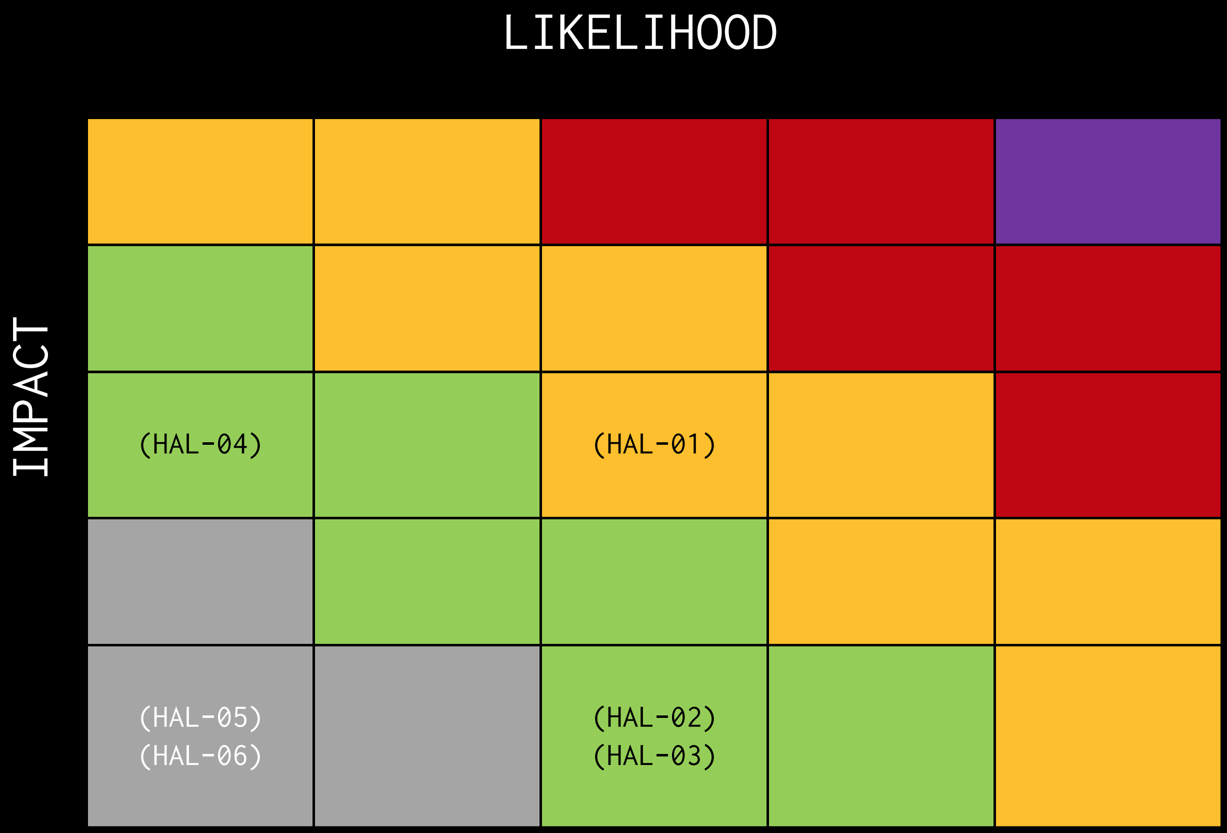
- Tracking module.
- Rewards module.

2. REMEDIATION PRs & COMMITS:

No commit/PR are provided by the Archway Team.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	1	3	2



🟨	🟨	🟥	🟥	🟪
🟩	🟨	🟨	🟥	🟥
🟩 (HAL-04)	🟩	🟨 (HAL-01)	🟨	🟥
⬜	🟩	🟩	🟨	🟨
⬜ (HAL-05) (HAL-06)	⬜	🟩 (HAL-02) (HAL-03)	🟩	🟨

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - MINIMUM CONSENSUS FEE NOT VALIDATED	Medium	RISK ACCEPTED
HAL-02 - SPECIFICATIONS DOCUMENTATION INCONSISTENCY	Low	RISK ACCEPTED
HAL-03 - DOCKER PRIVILEGED USER	Low	RISK ACCEPTED
HAL-04 - PANICS IN BEGINBLOCK AND ENDBLOCK	Low	RISK ACCEPTED
HAL-05 - PANIC IS USED FOR ERROR HANDLING	Informational	ACKNOWLEDGED
HAL-06 - OUTDATED OR VULNERABLE 3RD PARTY PACKAGES	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) MINIMUM CONSENSUS FEE NOT VALIDATED - MEDIUM

Description:

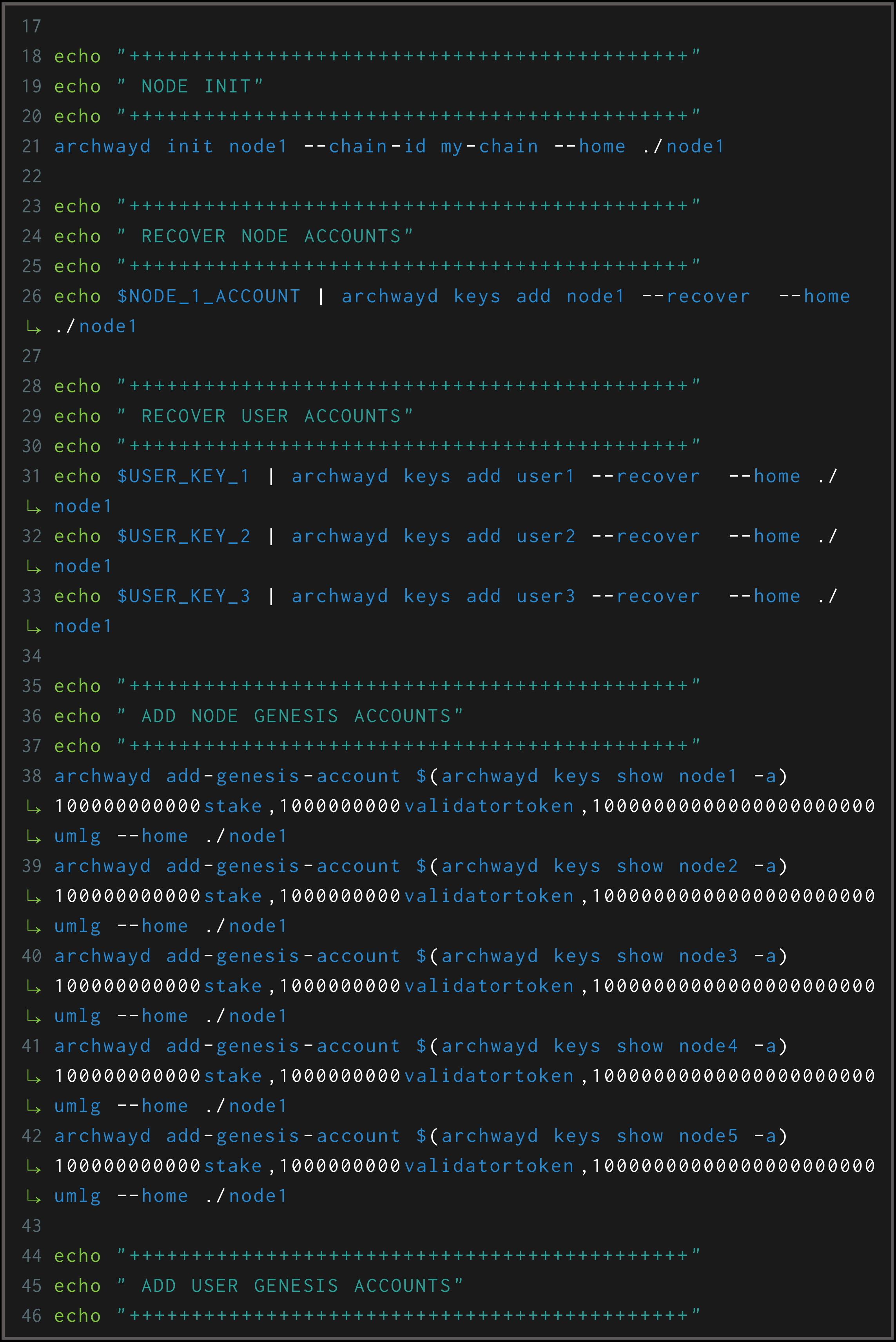
It was found that the MinConsensusFee parameter is not being validated to make sure that it is non-negative. By setting the value in genesis or by making a proposal, it could lead to a negative value fee.

This could lead to a malicious proposal being accepted and generating negative fees.

Proof of concept:

Listing 1

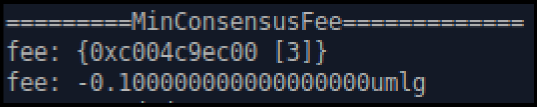
```
#!/bin/bash

cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway
rm -rf build
make build
export PATH=$PATH:`pwd`/build

rm -rf testnet/
mkdir testnet
cd testnet

export NODE_1_ACCOUNT="race draft rival universe maid cheese steel logic crowd fork comic easy truth drift tomorrow eye buddy head time cash swing swift midnight borrow"

export USER_KEY_1="hand inmate canvas head lunar naive increase recycle dog ecology inhale december wide bubble hockey dice worth gravity ketchup feed balance parent secret orchard"
export USER_KEY_2="alley afraid soup fall idea toss can goose become valve initial strong forward bright dish figure check leopard decide warfare hub unusual join cart"
export USER_KEY_3="record gift you once hip style during joke field prize dust unique length more pencil transfer quit train device arrive energy sort steak upset"
```



```
echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " NODE INIT"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
archwayd init node1 --chain-id my-chain --home ./node1

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " RECOVER NODE ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo $NODE_1_ACCOUNT | archwayd keys add node1 --recover  --home
↳ ./node1

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " RECOVER USER ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo $USER_KEY_1 | archwayd keys add user1 --recover  --home ./
↳ node1
echo $USER_KEY_2 | archwayd keys add user2 --recover  --home ./
↳ node1
echo $USER_KEY_3 | archwayd keys add user3 --recover  --home ./
↳ node1

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " ADD NODE GENESIS ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
archwayd add-genesis-account $(archwayd keys show node1 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node2 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node3 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node4 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node5 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node1

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " ADD USER GENESIS ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
```

```
archwayd add-genesis-account $(archwayd keys show user1 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show user2 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show user3 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1

echo "++++++++++++++++++++++++++++++++++++++++++++++"
echo " GENTX"
echo "++++++++++++++++++++++++++++++++++++++++++++++"
archwayd gentx node1 1000000000stake --chain-id my-chain --home ./node1

echo "++++++++++++++++++++++++++++++++++++++++++++++"
echo " COLLECT-GENTXS"
echo "++++++++++++++++++++++++++++++++++++++++++++++"
archwayd collect-gentxs --home ./node1

cat ./node1/config/genesis.json | jq '.app_state["rewards"]["min_consensus_fee"]["denom"]="umlg"' > /tmp/tmp_genesis.json && mv /tmp/tmp_genesis.json ./node1/config/genesis.json
cat ./node1/config/genesis.json | jq '.app_state["rewards"]["min_consensus_fee"]["amount"]="-0.1"' > /tmp/tmp_genesis.json && mv /tmp/tmp_genesis.json ./node1/config/genesis.json

echo "++++++++++++++++++++++++++++++++++++++++++++++"
echo " NODE IDS"
echo "++++++++++++++++++++++++++++++++++++++++++++++"
archwayd --home ./node1 tendermint show-node-id

export NODE1="$(archwayd --home ./node1 tendermint show-node-id)@127.0.0.1:26656"

perl -i -pe 's|"tcp://127.0.0.1:26657"|"tcp://127.0.0.1:26657"|g' ./node1/config/config.toml
perl -i -pe 's|"tcp://0.0.0.0:26656"|"tcp://127.0.0.1:26656"|g' ./node1/config/config.toml
perl -i -pe 's|"0.0.0.0:9090"|"127.0.0.1:9090"|g' ./node1/config/app.toml
perl -i -pe 's|"0.0.0.0:9091"|"127.0.0.1:9091"|g' ./node1/config/app.toml
perl -i -pe 's|"localhost:6060"|"127.0.0.1:6060"|g' ./node1/config/config.toml

clear; export PATH=$PATH:/home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/build; cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/testnet;archwayd --home ./node1 start

```

Screenshots:



```
=========MinConsensusFee=============
fee: {0xc004c9ec00 [3]}
fee: -0.100000000000000000umlg
```

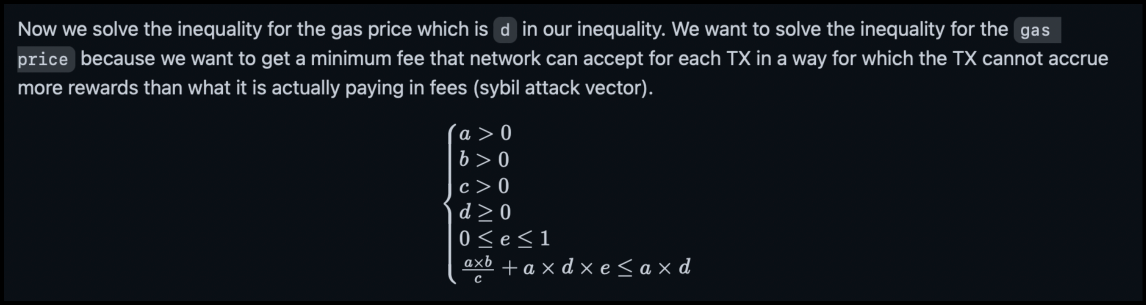
Figure 1: Debug message introduced to show negative fee

Recommendation:

It is recommended to implement a check to make sure the parameter is bigger than zero.

Remediation Plan:

RISK ACCEPTED: The Archway team accepted the risk of the issue. The issue raises concerns that the minimum consensus fee is negative, The Archway team claims that the formula is implemented in such a way that this would be impossible since d is great-than-or-equal to zero.



Now we solve the inequality for the gas price which is `d` in our inequality. We want to solve the inequality for the `gas price` because we want to get a minimum fee that network can accept for each TX in a way for which the TX cannot accrue more rewards than what it is actually paying in fees (sybil attack vector).

$$\begin{cases} a > 0 \\ b > 0 \\ c > 0 \\ d \geq 0 \\ 0 \leq e \leq 1 \\ \frac{a \times b}{c} + a \times d \times e \leq a \times d \end{cases}$$

Figure 2: Formula

3.2 (HAL-02) SPECIFICATIONS DOCUMENTATION INCONSISTENCY - LOW

Description:

It was found that in the x/rewards/spec/06_params.md specification document, the MaxWithdrawRecords parameter default value (1000) is inconsistent with what is being in action in the code. In the x/rewards/types/params.go file it gets set to 25000.

Code Location:

x/rewards/types/params.go, Lines 18-31

Listing 2: (Lines 21,30)

```
18 var (
19     // MaxWithdrawRecordsParamLimit defines the
 ↳ MaxWithdrawRecordsParamKey max value.
20     // Limit is estimated by the
 ↳ TestRewardsParamMaxWithdrawRecordsLimit E2E test.
21     MaxWithdrawRecordsParamLimit = uint64(25000) // limit is
 ↳ defined by the TestRewardsParamMaxWithdrawRecordsLimit E2E test
22     // MaxRecordsQueryLimit defines the page limit for querying
 ↳ RewardsRecords.
23     // Limit is defined by the TestRewardsRecordsQueryLimit E2E
 ↳ test.
24     MaxRecordsQueryLimit = uint64(7500)
25 )
26
27 var (
28     DefaultInflationRatio      = sdk.MustNewDecFromStr("0.20") //
 ↳ 20%
29     DefaultTxFeeRebateRatio    = sdk.MustNewDecFromStr("0.50") //
 ↳ 50%
30     DefaultMaxWithdrawRecords = MaxWithdrawRecordsParamLimit
31 )
```

Risk Level:

Likelihood - 3
Impact - 1

Recommendation:

It is recommended to keep the specifications document in line with the code to make sure that developers have accurate information.

Remediation Plan:

RISK ACCEPTED: The Archway team accepted the risk of the issue.

3.3 (HAL-03) DOCKER PRIVILEGED USER - LOW

Description:

It was found that the Dockerfile was insecurely configured. By not specifying a USER, the program inside the container may run as the root user. If an attacker can gain access to this container and control the process running as root, they may obtain control over the container.

Code Location:

Dockerfile, Line 49

Listing 3: (Lines 46,48)

```
46 ENTRYPOINT [ "/usr/bin/archwayd" ]
47
48 CMD [ "help" ]
```

Risk Level:

Likelihood - 3
Impact - 1

Recommendation:

It is recommended to specify a USER parameter that will point to a user with lower privileges.

Remediation Plan:

RISK ACCEPTED: The Archway team accepted the risk of the issue.

3.4 (HAL-04) PANICS IN BEGINBLOCK AND ENDBLOCK - LOW

Description:

BeginBlocker and EndBlocker are optional methods module developers can implement in their module.

They will be triggered at the beginning and at the end of each block, respectively, when the BeginBlock and EndBlock ABCI messages are received from the underlying consensus engine.

Making use of panics for error handling in the BeginBlock and EndBlock methods may cause the chain to halt if an error does occur. During the code review, It has been observed that If the chain zone does not have enough tokens, that can leads to chain halt.

Code Location:

x/rewards/abci.go, Line 18

Listing 4

```
18    k.AllocateBlockRewards(ctx, ctx.BlockHeight())
```

x/tracking/abci.go, Line 18

Listing 5

```
18    k.FinalizeBlockTxTracking(ctx)
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

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

Remediation Plan:

RISK ACCEPTED: The Archway team accepted the risk of the issue. The Archway team claims that they are aware that this constitutes correct / appropriate use of panic, as they represent unrecoverable errors that should result in a failed application execution or chain halts.

3.5 (HAL-05) PANIC IS USED FOR ERROR HANDLING - INFORMATIONAL

Description:

Several instances of the panic function were identified in the codebase. They appear to be used to handle errors. This can cause potential issues, as invoking a panic can cause the program to halt execution and crash in some cases. This in turn can negatively impact the availability of the software for users.

Code Location:

The following are just a few samples of the usage of panic:

Listing 6

```
./x/rewards/module.go:72:       panic(fmt.Errorf("registering query handler for x/%s: %w", types.ModuleName, err))
./x/rewards/mintbankkeeper/keeper.go:65:        panic(fmt.Errorf("unexpected dApp rewards: %s", dappRewards))
./x/rewards/keeper/state_tx_rewards.go:63:          panic(fmt.Errorf("invalid TxRewards Block index state: txId (%d): not found", txID))
./x/rewards/keeper/state_tx_rewards.go:170:     panic(fmt.Errorf("invalid TxRewards Block index key length: %d", len(key)))
./x/rewards/keeper/state_tx_rewards.go:175:     panic(fmt.Errorf("invalid TxRewards Block index key height: %d", heightRaw))
./x/rewards/keeper/withdraw.go:76:          panic(fmt.Errorf("sending rewards (%s) to the rewards address (%s): %w", totalRewards, rewardsAddr, err))
./x/rewards/keeper/state_metadata.go:79:        panic(fmt.Errorf("invalid contract address key: %w", err))
./x/rewards/keeper/distribution.go:260:     panic(fmt.Errorf("failed to transfer undistributed rewards (%s) to %s: %w", rewardsLeftovers, types.TreasuryCollector, err))
./x/rewards/keeper/state_rewards_record.go:74:          panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index state: id (%d): not found", id))
```

```
./x/rewards/keeper/state_rewards_record.go:95:          panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index state: id (%d): not found", id))
./x/rewards/keeper/state_rewards_record.go:199:     panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index key min length: %d", len(key)))
./x/rewards/keeper/state_rewards_record.go:205:     panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index key length: %d", len(key)))
./x/rewards/keeper/state_rewards_record.go:211:     panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index key (address): %s", err))
./x/rewards/keeper/state_rewards_record.go:223:     panic(fmt.Errorf("invalid RewardsRecord RewardsAddress index key min length (ID): %d", len(key)))
./x/rewards/types/rewards.go:94:        panic(fmt.Errorf("parsing rewardsRecord rewardsAddress: %w", err))
./x/rewards/types/msg.go:49:        panic(fmt.Errorf("parsing sender address (%s): %w", m.SenderAddress, err))
./x/rewards/types/msg.go:108:       panic(fmt.Errorf("parsing rewards address (%s): %w", m.RewardsAddress, err))
./x/rewards/types/events.go:15:     panic(fmt.Errorf("sending ContractMetadataSetEvent event: %w", err))
./x/rewards/types/events.go:28:     panic(fmt.Errorf("sending ContractRewardCalculationEvent event: %w", err))
./x/rewards/types/events.go:38:     panic(fmt.Errorf("sending RewardsWithdrawEvent event: %w", err))
./x/rewards/types/events.go:47:     panic(fmt.Errorf("sending MinConsensusFeeSetEvent event: %w", err))
./x/rewards/types/metadata.go:26:       panic(fmt.Errorf("parsing contract address: %w", err))
./x/rewards/types/metadata.go:37:       panic(fmt.Errorf("parsing rewards address (%s): %s", m.RewardsAddress, err))
./x/tracking/module.go:69:      panic(fmt.Errorf("registering query handler for x/%s: %w", types.ModuleName, err))
./x/tracking/keeper/state_tx_info.go:75:           panic(fmt.Errorf("invalid TxInfo Block index state: id (%d): not found", id))
./x/tracking/keeper/state_tx_info.go:191:       panic(fmt.Errorf("invalid TxInfo Block index key length: %d", len(key)))
./x/tracking/keeper/state_contract_op.go:66:            panic(fmt.Errorf("invalid ContractOpInfo TxInfo index state: id (%d): not found", id))
```

```
./x/tracking/keeper/state_contract_op.go:191:       panic(fmt.Errorf("invalid ContractOpInfo TxInfo index key length: %d", len(key)))
./x/tracking/types/tracking.go:41:      panic(fmt.Errorf("parsing contract address (%s): %w", m.ContractAddress, err))
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

Remediation Plan:

ACKNOWLEDGED: The Archway team acknowledged this issue.

3.6 (HAL-06) OUTDATED OR VULNERABLE 3RD PARTY PACKAGES - INFORMATIONAL

Description:

Outdated 3rd party packages were in use. The outdated packages as well as known vulnerabilities for these packages are listed below.

ID	Package	Rating	Description
sonatype-2021-0598	tendermint	MEDIUM	Improper Input Validation
CVE-2022-32149	text	HIGH	Resource Exhaustion
sonatype-2021-0456	websocket	HIGH	Resource Exhaustion

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended that any 3rd party package or module is always kept up to date, or the latest security patches applied.

Remediation Plan:

ACKNOWLEDGED: The Archway team acknowledged this issue.

AUTOMATED TESTING

Description:

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

Semgrep - Security Analysis Output Sample:

Listing 7: Rule Set

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

Semgrep Results:

Listing 8

```
Findings:

  Dockerfile
     dockerfile.security.missing-user.missing-user
        By not specifying a USER, a program in the container may run as 'root'. This is a security
        hazard. If an attacker can control a process running as root, they may have control over the
        container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.
        Details: https://sg.run/Gbvn

         46 ENTRYPOINT [ "/usr/bin/archwayd" ]
          ----------------------------------------
         48 CMD [ "help" ]


  ci/constantine-1-bigdipper.yaml
     yaml.kubernetes.security.allow-privilege-escalation.allow-privilege-escalation
        Container constantine-1-bigdipper allows for privilege escalation via setuid or setgid
        binaries. Add 'allowPrivilegeEscalation: false' in 'securityContext' to prevent this.
        Details: https://sg.run/ljp6

        107 - name: constantine-1-bigdipper
          ----------------------------------------
     yaml.kubernetes.security.run-as-non-root.run-as-non-root
        Container allows for running applications as root. This can result in privilege escalation
        attacks. Add 'runAsNonRoot: true' in 'securityContext' to prevent this.
        Details: https://sg.run/dgP5

        107 - name: constantine-1-bigdipper


  ci/titus-1-bigdigger.yaml
     yaml.kubernetes.security.allow-privilege-escalation.allow-privilege-escalation
        Container titus-1-bigdipper allows for privilege escalation via setuid or setgid binaries.
        Add 'allowPrivilegeEscalation: false' in 'securityContext' to prevent this.
        Details: https://sg.run/ljp6

        107 - name: titus-1-bigdipper
          ----------------------------------------
     yaml.kubernetes.security.run-as-non-root.run-as-non-root
```

```
        Container allows for running applications as root. This can result in privilege escalation
        attacks. Add 'runAsNonRoot: true' in 'securityContext' to prevent this.
        Details: https://sg.run/dgP5

        107 - name: titus-1-bigdipper


  ci/torii-1-bigdipper.yaml
     yaml.kubernetes.security.allow-privilege-escalation.allow-privilege-escalation
        Container torii-1-bigdipper allows for privilege escalation via setuid or setgid binaries.
        Add 'allowPrivilegeEscalation: false' in 'securityContext' to prevent this.
        Details: https://sg.run/ljp6

        114 name: torii-1-bigdipper
          ----------------------------------------
     yaml.kubernetes.security.run-as-non-root.run-as-non-root
        Container allows for running applications as root. This can result in privilege escalation
        attacks. Add 'runAsNonRoot: true' in 'securityContext' to prevent this.
        Details: https://sg.run/dgP5

        114 name: torii-1-bigdipper


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

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

          4 node:
```

Gosec - Security Analysis Output Sample:

Listing 9

```
[x/tracking/keeper/state_contract_op.go:101] - G601 (CWE-118): Implicit memory aliasing in for loop. (Confidence: MEDIUM, Severity: MEDIUM)
    100:     for _, obj := range objs {
  > 101:         s.setContractOpInfo(&obj)
    102:         s.setTxIndex(obj.TxId, obj.Id)



[x/rewards/keeper/state_tx_rewards.go:74] - G601 (CWE-118): Implicit memory aliasing in for loop. (Confidence: MEDIUM, Severity: MEDIUM)
    73:      for _, obj := range objs {
  > 74:          s.setTxRewards(&obj)
    75:          s.setBlockIndex(obj.Height, obj.TxId)



[x/rewards/keeper/state_rewards_record.go:119] - G601 (CWE-118): Implicit memory aliasing in for loop. (Confidence: MEDIUM, Severity: MEDIUM)
    118:     for _, obj := range objs {
  > 119:         s.setRewardsRecord(&obj)
    120:         s.setAddressIndex(obj.Id, obj.MustGetRewardsAddress())



[x/rewards/keeper/state_block_rewards.go:56] - G601 (CWE-118): Implicit memory aliasing in for loop. (Confidence: MEDIUM, Severity: MEDIUM)
    55:      for _, obj := range objs {
  > 56:          s.setBlockRewards(&obj)
    57:      }

```

Staticcheck - Security Analysis Output Sample:

Listing 10

```
app/test_access.go:22:2: field t is unused (U1000)
e2e/testing/chain.go:25:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
e2e/testing/chain.go:64:10: assigning the result of this type assertion to a variable (switch opt := opt.(type)) could eliminate type assertions in switch cases (S1034)
	e2e/testing/chain.go:66:40: could eliminate this type assertion
	e2e/testing/chain.go:68:54: could eliminate this type assertion
	e2e/testing/chain.go:70:40: could eliminate this type assertion
e2e/testing/ibc_path.go:189:23: func (*IBCEndpoint).sendPacket is unused (U1000)
x/rewards/keeper/state_metadata.go:76:32: func ContractMetadataState.parseContractMetadataKey is unused (U1000)
x/rewards/module.go:92:2: field cdc is unused (U1000)
x/rewards/types/query.pb.gw.go:17:2: "github.com/golang/protobuf/descriptor" is deprecated: See the "google.golang.org/protobuf/reflect/protoreflect" package for how to obtain an EnumDescriptor or MessageDescriptor in order to programatically interact with the protobuf type system.  (SA1019)
x/rewards/types/query.pb.gw.go:18:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/rewards/types/query.pb.gw.go:33:9: descriptor.ForMessage is deprecated: Not all concrete message types satisfy the Message interface. Use MessageDescriptorProto instead. If possible, the calling code should be rewritten to use protobuf reflection instead. See package "google.golang.org/protobuf/reflect/protoreflect" for details.  (SA1019)
x/tracking/keeper/keeper.go:20:2: field paramStore is unused (U1000)
x/tracking/module.go:89:2: field cdc is unused (U1000)
x/tracking/types/query.pb.gw.go:17:2: "github.com/golang/protobuf/descriptor" is deprecated: See the "google.golang.org/protobuf/reflect/protoreflect" package for how to obtain an EnumDescriptor or MessageDescriptor in order to programatically interact with the protobuf type system.  (SA1019)
```

```
x/tracking/types/query.pb.gw.go:18:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/tracking/types/query.pb.gw.go:33:9: descriptor.ForMessage is deprecated: Not all concrete message types satisfy the Message interface. Use MessageDescriptorProto instead. If possible, the calling code should be rewritten to use protobuf reflection instead. See package "google.golang.org/protobuf/reflect/protoreflect" for details.  (SA1019)
```

THANK YOU FOR CHOOSING

HALBORN