// HALBORN

SifChain - 0.42 Update

Security Audit

Prepared by: **Halborn**

Date of Engagement: **June 12, 2021 - August 12, 2021**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	07/12/2021	Gokberk Gulgun
0.2	Document Updates	07/13/2021	Gokberk Gulgun
0.3	Add Admin Findings	07/14/2021	Todd Garrison
0.4	Revise High and Medium Findings	07/15/2021	Todd Garrison
0.5	Add Contracts Ownership Review	07/15/2021	Todd Garrison
1.0	Final Review	07/16/2021	Steven Walbroehl
1.1	Remediation Plan	08/03/2021	Gokberk Gulgun
1.1	Remediation Review	08/04/2021	Steven Walbroehl
1.2	Remediation Final Draft	08/17/2021	Steven Walbroehl

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Todd Garrison	Halborn	Todd.Garrison@halborn.com
Gokberk Gulgun	Halborn	Gokberk.Gulgun@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Sifchain engaged Halborn to conduct a security assessment on their repository beginning on June 12th, 2021 and ending August 12th, 2021. This security assessment was scoped to the 0.42 update. Halborn was provided access to the source code of the application and the testing environment in order to conduct security testing using tools to scan, detect, validate possible vulnerabilities found in the application and report the findings at the end of the engagement.

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks for the engagement and assigned three full time security engineers to audit the security of the smart contract. The security engineers are blockchain and smart-contract security experts 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 Sifnode's functionalities are intended.
- Identify potential security issues with the Sifnode structure.

In summary, Halborn identified few security vulnerabilities, but does have concerns around centralized control of the application. Halborn recommends performing further testing to validate extended safety and correctness in context to the whole structure.

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 structures. 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 structures, and imported functions. (gosec, shadow, staticcheck, errcheck)
- Manual Assessment for discovering security vulnerabilities on code-bases.
- Review of governance and centralization risks.
- Dynamic Analysis

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. It's quantitative model ensures repeatable and accurate measurement

while enabling users to see the underlying vulnerability characteristics that was 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.

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

1.4 SCOPE

IN-SCOPE : 0.42 Update - https://github.com/Sifchain/sifnode/pull/1513/files

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	3	5	2

LIKELIHOOD

IMPACT

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

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - PRIVILEGED ACCOUNT CAN ACCESS PEGGED FUNDS	Medium	ACCEPTED RISK - 08/04/2021
HAL02 - ADMIN ACCOUNT IS HARD-CODED AT GENESIS	Low	ACCEPTED RISK - 08/04/2021
HAL03 - ONLY ADMINS CAN ASSIGN ORACLES	Medium	ACCEPTED RISK - 08/04/2021
HAL04 - NETWORK CENTRALIZATION	Low	ACCEPTED RISK - 08/04/2021
HAL05 - ORACLES SIGN WITH VALIDATOR KEYS	Low	ACCEPTED RISK - 08/04/2021
HAL06 - SENSITIVE INFORMATION IN ENVIRONMENT VARIABLES	Low	FUTURE RELEASE UPDATE
HAL07 - HARDCODED MNEMONIC PHRASES IN THE REPOSITORY	Low	ACCEPTED RISK - 08/04/2021
HAL08 - UNBOUNDED LOOPS ON THE FUNCTIONS	Informational	FUTURE RELEASE UPDATE
HAL09 - RUN CONTAINER AS ROOT	Low	FUTURE RELEASE UPDATE
HAL10 - UNSAFE RPC EXPOSED	Informational	FUTURE RELEASE UPDATE

FINDINGS & TECH DETAILS

3.1 (HAL-01) PRIVILEGED ACCOUNT CAN ACCESS PEGGED FUNDS - MEDIUM

Description:

The primary function of Sifchain is to offer a cross-chain decentralized exchange. This is achieved by using a decentralized group of "oracles" that have special privileges. The Peggy design document states that the oracles should use a consensus model. And is used for cross-chain swaps on both the Ethereum and Sif networks, however a single "Admin" account is capable of withdrawing tokens via the RescueCeth functionality. This is without consensus or governance, and is entirely unilateral. The risk is compounded by additional findings outlined in this report.

Code Location:

IsAdminAccount is the function used to authenticate the Admin.

Listing 1: x/oracle/keeper/adminAccount.go

```
18 func (k Keeper) IsAdminAccount(ctx sdk.Context, adminAccount sdk.
       AccAddress) bool {
19     account := k.GetAdminAccount(ctx)
20     if account == nil {
21         return false
22     }
23     return bytes.Equal(account, adminAccount)
24 }
```

This is used as the only authority check in the ProcessRescueCeth and ProcessUpdateCethReceiverAccount function, if the following test passes, all ceth (aka wrapped ethereum assets on Sifchain) can be withdrawn to an arbitrary Sifchain account.

Listing 2: x/oracle/keeper/keeper.go

```
215     if !k.oracleKeeper.IsAdminAccount(ctx, cosmosSender) {
216         logger.Error("cosmos sender is not admin account.")
```

```
217         return errors.New("only admin account can call rescue ceth
                ")
218     }
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendations:

Short Term:

- Setup monitoring for activity on the Admin account, any activity should immediately trigger a response.
- Prepare a “break-glass” procedure for the case where this account is compromised. What needs to be done on both chains, who will do it, how to communicate the issue, community interaction. All incident response tasks need to be pre-planned. Assume that it will happen, and design the response to minimize the damage. This procedure will likely involve pausing contracts and halting the oracle nodes to prevent ongoing damage.

Long Term:

- Eliminate the Admin account.
- Remove any privileged account functions and move these abilities as a function of governance or delegate the functionality to a super majority vote of either oracles or validators.

Remediation Plan:

ACKNOWLEDGED: Sifchain Team claims that `ProcessRescueCeth` and `ProcessUpdateCethRecieverAccount` do not have access to all the pegged

funds deposited by users. They can only access cETH paid by users as fees to export their assets to Ethereum. ProcessRescueCeth1 and ProcessUpdateCethRecieverAccount do NOT have access to all the pegged funds deposited by users. They can only access cETH paid by users as fees to export their assets to Ethereum. When users export assets from Sifchain to Ethereum, they pay cETH to a Receiver Account that represents the Relayers. The relayers are on both Ethereum and Sifchain, only they can accept cETH on Ethereum and pay a corresponding amount of ETH on Ethereum on behalf of the user to send them their exported Ethereum assets. The purpose of ProcessRescueCeth and ProcessUpdateCethRecieverAccount is to occasionally gather all the cETH paid by users as fees, have the admin export it to Ethereum, and then give it to the Relayer so it can pay additional fees.

3.2 (HAL-02) ADMIN ACCOUNT IS HARD-CODED AT GENESIS - LOW

Description:

The Admin account is defined at genesis and there does not seem to be a mechanism to update the account. If the key is compromised it would be impossible to contain the attack without stopping the chain and deploying new code. This further amplifies the risk of HAL-01 because there is no containment mechanism in the case of a compromised key, and the only resulting possible solution is to halt all oracles and pause contracts to prevent fund transfers, assuming the funds have not already been moved.

Code Location:

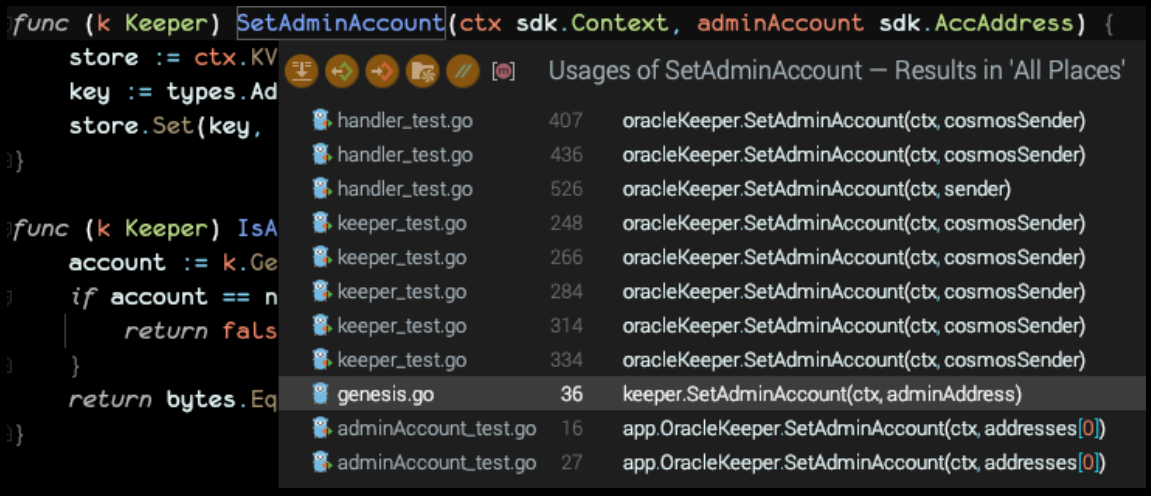


Figure 1: Only reference to SetAdminAccount

The account being allocated can be seen in the genesis.json file:

Listing 3: .sifnoded/config/genesis.json

```
193     "oracle": {
194       "address_whitelist": null,
195       "admin_address": "sif1wy22rv4w3l9vmw5cvdc6npg8t0vess0kpvga3j
            "
196     },
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendations:

Eliminate the Admin role and use governance.

Remediation Plan:

ACCEPTED RISK: The risk has been taken by the Sifchain Team. There are methods to upgrade the Genesis with a Network Upgrade.

3.3 (HAL-03) ONLY ADMINS CAN ASSIGN ORACLES - MEDIUM

Code Location:

Listing 4: x/oracle/keeper/keeper.go

```
178 // ProcessUpdateWhiteListValidator processes the update whitelist
      validator from admin
179 func (k Keeper) ProcessUpdateWhiteListValidator(ctx sdk.Context,
      cosmosSender sdk.AccAddress, validator sdk.ValAddress,
      operationtype string) error {
180     logger := k.Logger(ctx)
181     if !k.IsAdminAccount(ctx, cosmosSender) {
182         logger.Error("cosmos sender is not admin account.")
183         return types.ErrNotAdminAccount
184     }
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendations:

The process of electing an oracle should be handled by validators, and validators should have the option of running an oracle with some sort of financial incentive.

Remediation Plan:

ACCEPTED RISK: Sifchain Team claims that oracle assignment can be completed by the admin. However, there are future plans to further decentralize the Assignment of Oracles with certain services, such as MultiSignature solutions.

3.4 (HAL-04) NETWORK CENTRALIZATION - LOW

Description:

Sifchain has stated that decentralization is an eventual goal, with a promise of transitioning to a DAO with full on-chain governance post-Betanet. However, the chain has millions of dollars of assets pegged right now. The network validator, oracle, and admin roles are almost entirely controlled by Sifchain.

Eighty-six percent of the network's validator stake is controlled by Sifchain and assigned to nodes run by Sifchain. This not only gives the appearance of a centralized system, it also disincentives smaller validators from running nodes further contributing to the problem.

It also appears that this centralization has been downplayed by modifying the BigDipper-based Sifchain block explorer to show a set of random validators where recently signed blocks would normally appear, giving the illusion of a more decentralized validator set.

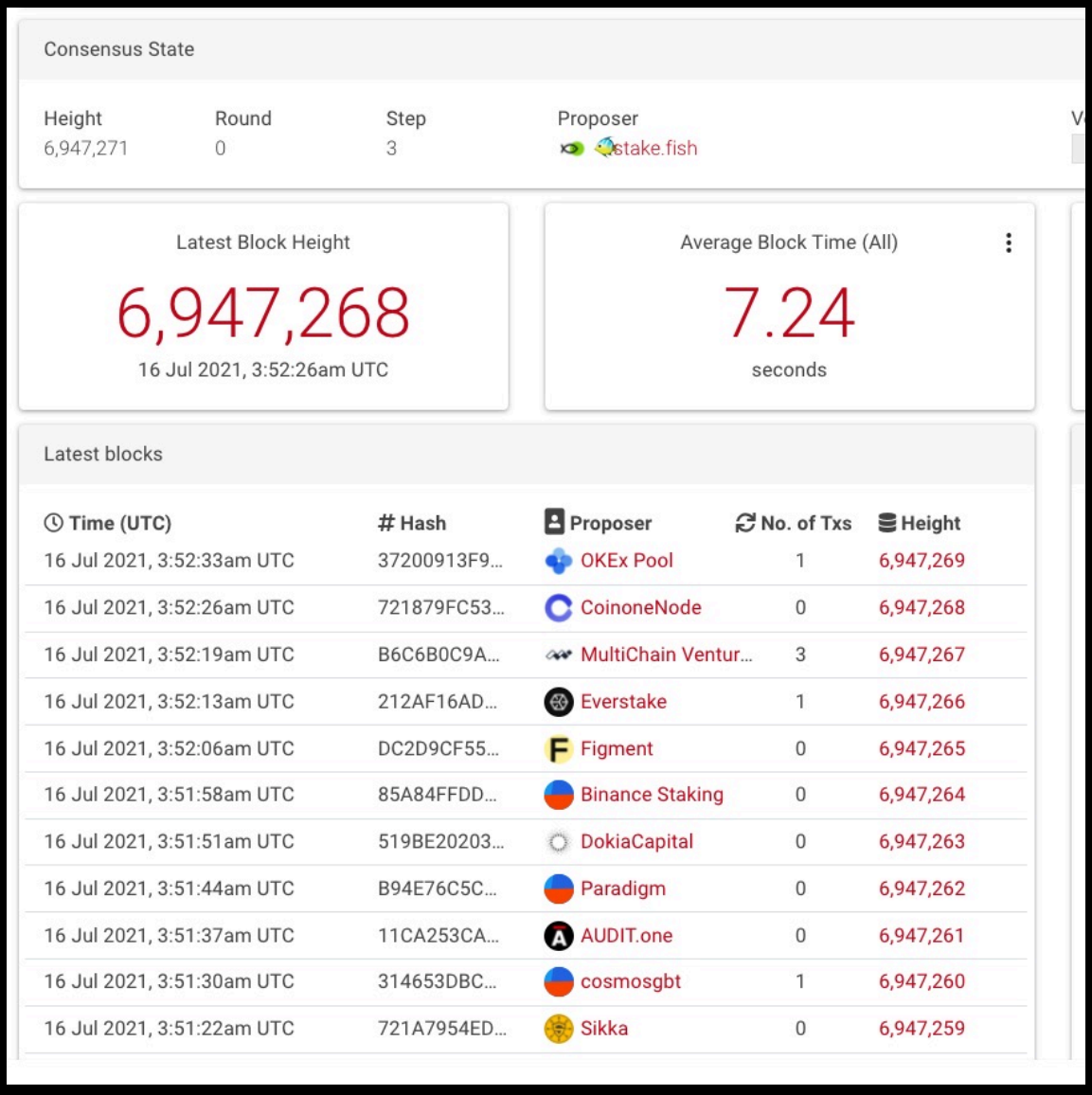


Figure 2: Cosmos Big Dipper Shows Recent Blocks

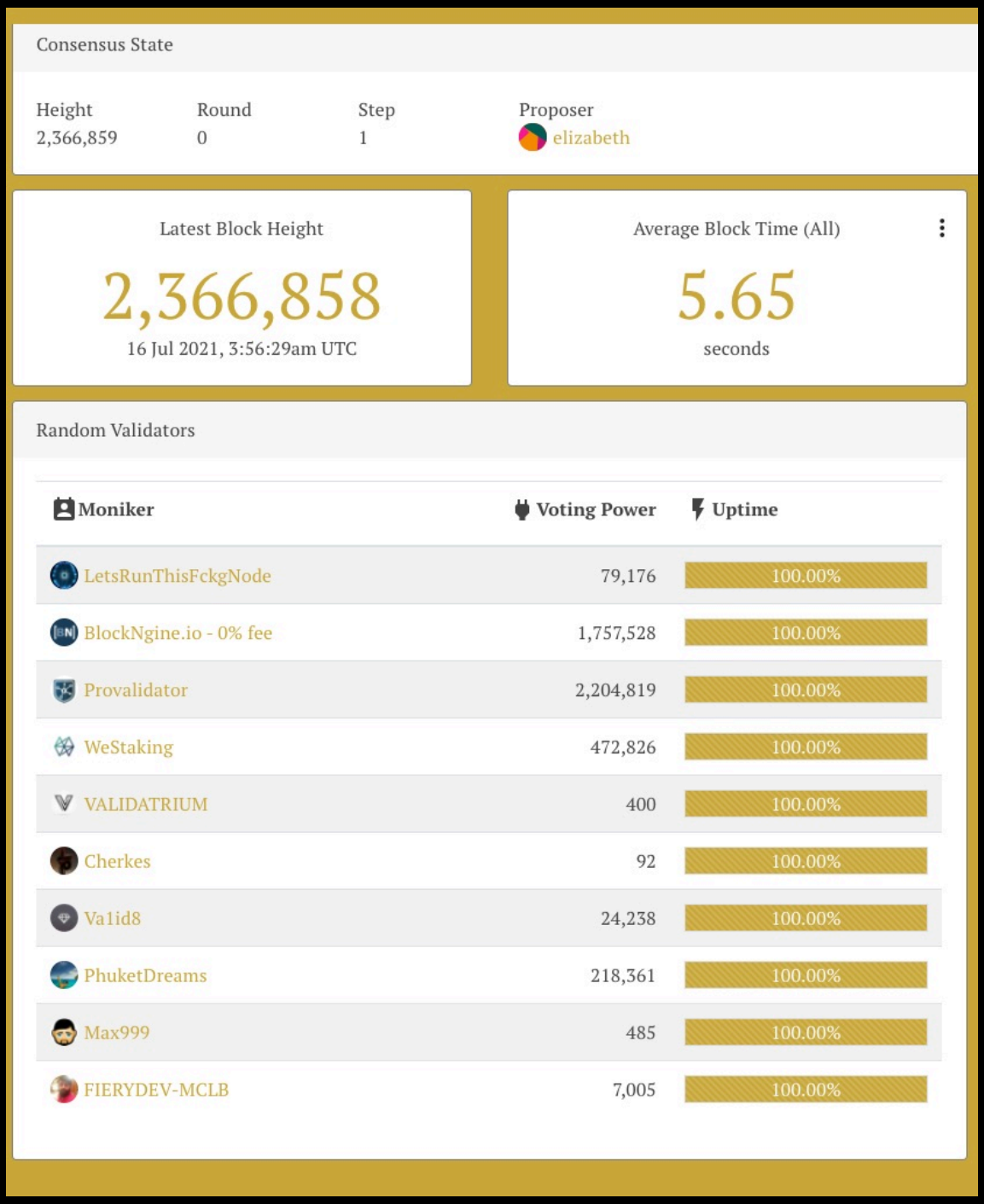


Figure 3: Sifchain shows random validators instead

Risk Level:

Likelihood - 2
Impact - 2

Recommendations:

Despite running on a public blockchain and inviting validators to participate, validators and token holders have little power. The last governance proposals were immediately and solely approved by Sifchain-owned accounts.

- Sifchain should reduce the amount of rowan staked to their validators. This could be done by transitioning to an incentive model, such as the Kusama Thousand Validator program, or even the Osmosis foundation's delegation program, where funds are allocated to validators that fulfill certain requirements for uptime, governance participation or telemetry statistics.
- There should be a minimum stake required for validators that is high enough to ensure quality validators are securing the network.
- The size of the validator pool should be reduced if there are not sufficient validators to create competition for joining the active set.

Remediation Plan:

PARTIALLY FIXED:

Sifchain retired the old block explorer entirely in favor of this 3rd party block explorer run by Cosmostation https://www.mintscan.io/sifchain. Sifchain's foundation validators cannot earn any validator income as it's all redistributed to the other validators, so the disincentivization issue is not an issue. That said, we agree the network's security comes from the foundation's validators.
Sifchain is considering a foundation delegation program with the following constraints:

1. Known validator that can sign a contract
2. Minimum self-delegation of some number of Rowan (not sure the amount yet)
3. Minimum 2% commission, maximum 10% commission
4. Not being slashed for not updating to a new node.
5. Bonus points based on content in Persistence’s program

3.5 (HAL-05) ORACLES SIGN WITH VALIDATOR KEYS - LOW

Description:

The Oracle role is required to be a validator. While this may seem like a way to further restrict permissions it does not provide any additional protection. Sifchain only has seventy two active producers and the lowest self-bond is 1rowan which at the time of this report is $0.12 USD. Using the same key on both a Validator and Oracle increases the risk of the Oracle key being disclosed, especially if they are run on separate nodes. This also violates the principle of least privilege, and given the power Oracles wield reducing exposure should be a priority.

Code Location:

The following check enforces this behavior:

Listing 5: x/staking/keeper/validator.go

```
26 func (k Keeper) mustGetValidator(ctx sdk.Context, addr sdk.
       ValAddress) types.Validator {
27     validator, found := k.GetValidator(ctx, addr)
28     if !found {
29         panic(fmt.Sprintf("validator record not found for address:
               %X\n", addr))
30     }
31
32     return validator
33 }
```

Risk Level:

Likelihood - 2
Impact - 3

Recommendations:

Authorization for oracle or bridge related actions and validators should use different roles.

Remediation Plan:

PENDING: Sifchain Team will fix it in a future release.

3.6 (HAL-06) SENSITIVE INFORMATION IN ENVIRONMENT VARIABLES - MEDIUM

Description:

When you store your secret keys in an environment variable, you are prone to accidentally exposing them.

Code Location:

Signature Go File

Listing 6: cmd/ebrelayer/txs/signature.go

```
21 func LoadPrivateKey() (key *ecdsa.PrivateKey, err error) {
22     // Private key for validator's Ethereum address must be set as
           an environment variable
23     rawPrivateKey := os.Getenv("ETHEREUM_PRIVATE_KEY")
24     ....
```

Risk Level:

Likelihood - 3
Impact - 3

Recommendations:

Given that the environment is implicitly available to the process, it's hard, if not impossible, to track access and how the contents get exposed (ps -eww).
It's common to have applications grab the whole environment and print it out for debugging or error reporting. Environment variables are passed down to child processes, which allows for unintended access. This breaks the principle of least privilege. Imagine that as part of your application, you call to a third-party tool to perform some action---all of a sudden that third-party tool has access to your

environment. When applications crash, it’s common for them to store the environment variables in log-files for later debugging. This means plain-text secrets on disk.

Remediation Plan:

PENDING: Sifchain Team will fix it in a future release with a refactor of using options such as a Key Manager external to the environment such as HSM or Secrets Manager. They need to use an HSM to generate keys and AWS Secrets Manager to retrieve to address 06

3.7 (HAL-07) HARDCODED MNEMONIC PHRASES IN THE REPOSITORY - LOW

Description:

In the SifNode repository, it was discovered that there are multiple mnemonic phrases hardcoded in bash scripts.

At least one of the keys have been in active use on both the Sifchain and Ethereum mainnets, in particular the ‘sif’ account referenced below.

- Sifchain Transactions
- Ethereum Transactions

Code Location:

init.sh

Listing 7: /scripts/init.sh (Lines)

```
1 echo "Generating deterministic account - sif"
2 echo "race draft rival universe maid cheese steel logic crowd fork
      comic easy truth drift tomorrow eye buddy head time cash swing
      swift midnight borrow" | sifnodecli keys add sif --recover
3
4 echo "Generating deterministic account - akasha"
5 echo "hand inmate canvas head lunar naive increase recycle dog
     ecology inhale december wide bubble hockey dice worth gravity
     ketchup feed balance parent secret orchard" | sifnodecli keys
     add akasha --recover
6
```

sifchain_start_daemon.sh

Listing 8: sifchain start daemon.sh (Lines)

```
1 #{"name":"fnord","type":"local","address":"
     sif10ckfjtdmk9zkcs9fhl0h260xsj6kvg7esmyqrw","pubkey":"","
```

```
      mnemonic":"exact below syrup slender party witness already lamp
       inform dash impose ginger sauce shift tag humble awkward spawn
       blue flower lab census gold girl"}
2
```

Risk Level:

Likelihood - 2
Impact - 3

Recommendations:

Use secure vault to store credentials instead of using them hardcoded. Also, It is recommended to delete mnemonic phrases from the Git history, and change any previously used mnemonic phrases to prevent future incidents if these mnemonic phrases have been leaked, or reused elsewhere. Test scripts should dynamically create keys at runtime.

Remediation Plan:

ACCEPTED RISK: Sifchain Team claims that the mnemonic phrases used only for test purpose.

3.8 (HAL-08) UNBOUNDED LOOPS ON THE FUNCTIONS - INFORMATIONAL

Description:

peggyTokens and distributionRecordsclass contain an iteration which does not have bound on the itself. If the loop can be influenced by an attacker, this weakness could allow attackers to consume excessive resources such as CPU or memory.The program contains an iteration or loop with an exit condition that cannot be reached.

Code Location:

peggyTokens.go

Listing 9: x/ethbridge/keeper/peggyTokens.go (Lines)

```
23 ....
24     tokens := k.GetPeggyToken(ctx)
25     for _, value := range tokens {
26         if value == token {
27             return true
28         }
29     }
30     return false
31 ...
```

distributionRecords.go

Listing 10: https://github.com/Sifchain/sifnode/blob/d1ecf5449b7ffae9ae279fb1246844 (Lines)

```
214 func (k Keeper) GetRecords(ctx sdk.Context) types.
       DistributionRecords {
215     var res types.DistributionRecords
216     iterator := k.GetDistributionRecordsIterator(ctx)
217     defer iterator.Close()
218     for ; iterator.Valid(); iterator.Next() {
219         var dr types.DistributionRecord
```

```
220         bytesValue := iterator.Value()
221         k.cdc.MustUnmarshalBinaryBare(bytesValue, &dr)
222         res = append(res, dr)
223     }
224     return res
225 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendations:

Consider implementing bounds on the functions.

Remediation Plan:

PENDING: Sifchain Team will fix it in a future release.

3.9 (HAL-09) RUN CONTAINER AS ROOT - LOW

Description:

Configuring the container to use an unprivileged user is the best way to prevent privilege escalation attacks. It is possible to run sifnode as a root via Docker.

Sample Docker Image:

Location: https://github.com/Sifchain/sifnode/blob/9597894b9628ea5eebf41fd2ef7c7533f4022be2/deploy/docker/mainnet/docker-compose.yml#L6

Listing 11

```
1 FROM sifchain/sifnoded:mainnet-genesis
2 RUN addgroup -g 26656 sif && adduser -u 26656 -G sif -S -h /app
      sif && \
3       mkdir -p /app/.sifnoded/cosmovisor/genesis/bin /app/.sifnoded
           /cosmovisor/upgrades \
4         /app/.sifnoded/config /app/.sifnoded/data && \
5       mv /usr/bin/sifnoded /app/.sifnoded/cosmovisor/genesis/bin/
           && chown -R sif:sif /app
6 USER sif
7 WORKDIR /app
8 ENV PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
      bin:/app/.sifnoded/cosmovisor/genesis/bin
9
```

Risk Level:

Likelihood - 3
Impact - 2

Recommendations:

One of the best practices while running Docker Container is to run processes with a non-root user. This is because if a user manages to break out of the application running as root in the container, an attacker may gain root user access on host.

Remediation Plan:

PENDING: Sifchain Team will fix it in a future release.

3.10 (HAL-10) UNSAFE RPC EXPOSED - INFORMATIONAL

Description:

Sifnode Docker image(docker-compose) exposes RPC by default instead of binding to localhost.

Code Location:

Example RPC Port Definition

Listing 12

```
1 TCP_URL=tcp://0.0.0.0:26657
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendations:

Ensure that the RPC port is not exposed to 0.0.0.0. When the RPC exposed, validators are vulnerable to a possible DDOS attack.

Remediation Plan:

PENDING: Sifchain Team will fix it in a future release.

STATIC ANALYSIS REPORT

4.1 STATIC ANALYSIS

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped SifNode repository. Among the tools used were staticcheck, gosec errcheck and others. 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.

Gosec - Security Analysis Output Sample:

```
[/home/destek/Downloads/sifnode/x/clp/test/test_common.go:91] - G404 (CWE-338): Use of weak random number generator (math/rand instead of crypto/rand) (Confidence: MEDIUM, Severity: HIGH)
   90:         for i := 0; i < numberOfLp; i++ {
  > 91:                 externalToken := tokens[rand.Intn(len(tokens))]
   92:                 asset := types.NewAsset(trimFirstRune(externalToken))


[/home/destek/Downloads/sifnode/x/clp/test/test_common.go:75] - G404 (CWE-338): Use of weak random number generator (math/rand instead of crypto/rand) (Confidence: MEDIUM, Severity: HIGH)
   74:                 // inittalize global pseudo random generator
  > 75:                 externalToken := tokens[rand.Intn(len(tokens))]
   76:                 externalAsset := types.NewAsset(trimFirstRune(externalToken))


[/home/destek/Downloads/sifnode/tools/sifgen/utils/cli.go:235] - G204 (CWE-78): Subprocess launched with variable (Confidence: HIGH, Severity: MEDIUM)
   238: func (c CLI) shellExecInput(cmd string, inputs [][]byte, args ...string) (*string, error) {
  > 239:         cm := exec.Command(cmd, args...)
   240:         var stderr bytes.Buffer


[/home/destek/Downloads/sifnode/tools/sifgen/utils/cli.go:225] - G204 (CWE-78): Subprocess launched with variable (Confidence: HIGH, Severity: MEDIUM)
   224: func (c CLI) shellExec(cmd string, args ...string) (*string, error) {
  > 225:         cm := exec.Command(cmd, args...)
   226:         var out bytes.Buffer


[/home/destek/Downloads/sifnode/cmd/ebrelayer/contract/generator.go:74] - G204 (CWE-78): Subprocess launched with variable (Confidence: HIGH, Severity: MEDIUM)
   73: func execCmd(cmd string) error {
  > 74:         _, err := exec.Command("sh", "-c", cmd).Output()
   75:         return err


[/home/destek/Downloads/sifnode/x/dispensation/utils/parser.go:43] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   42:         }
  > 43:         o, err := ioutil.ReadFile(file)
   44:         if err != nil {


[/home/destek/Downloads/sifnode/x/dispensation/utils/parser.go:25] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   24:         }
  > 25:         input, err := ioutil.ReadFile(file)
   26:         if err != nil {
```

```
[/home/destek/Downloads/sifnode/tools/sifgen/network/network.go:371] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   370:                 if !validator.Seed {
  > 371:                         input, err := ioutil.ReadFile(srcFile)
   372:                         if err != nil {


[/home/destek/Downloads/sifnode/tools/sifgen/network/network.go:338] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   337:
  > 338:                 content, err := ioutil.ReadFile(configFile)
   339:                 if err != nil {


[/home/destek/Downloads/sifnode/tools/sifgen/genesis/genesis.go:152] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   151:
  > 152:         body, err := ioutil.ReadFile(genesisPath)
   153:         if err != nil {


[/home/destek/Downloads/sifnode/cmd/ebrelayer/contract/abi.go:41] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   40:         // Read the file containing the contract's ABI
  > 41:         contractRaw, err := ioutil.ReadFile(dir + filePath)
   42:         if err != nil {


[/home/destek/Downloads/sifnode/tools/sifgen/utils/cli.go:89] - G301 (CWE-276): Expect directory permissions to be 0750 or less (Confidence: HIGH, Severity: MEDIUM)
   88: func (c CLI) CreateDir(path string) error {
  > 89:         return os.MkdirAll(path, 0755)
   90: }
```

Staticcheck - Security Analysis Output Sample:

```
cmd/ebrelayer/relayer/cosmos.go:227:7: should use !bytes.Equal(tx.Data()[0:4], methodID) instead (S1004)
cmd/ebrelayer/relayer/cosmos.go:272:6: should use bytes.Equal(message.CosmosSender, prophecyClaim.CosmosSender) instead (S1004)
cmd/ebrelayer/relayer/ethereum.go:404:16: unnecessary use of fmt.Sprintf (S1039)
x/clp/genesis.go:14:12: unnecessary use of fmt.Sprintf (S1039)
x/clp/keeper/pool.go:26:2: should use 'return err == nil' instead of 'if err != nil { return false }; return true' (S1008)
x/clp/types/types.go:24:2: should use 'return p.ExternalAsset.Validate()' instead of 'if !p.ExternalAsset.Validate() { return false }; return true' (S1008)
x/clp/types/types.go:57:2: should use 'return l.Asset.Validate()' instead of 'if !l.Asset.Validate() { return false }; return true' (S1008)
x/dispensation/keeper/distribution.go:35:2: should use 'return k.Exists(ctx, key)' instead of 'if k.Exists(ctx, key) { return true }; return false' (S1008)
x/dispensation/keeper/distributionRecords.go:61:2: should use 'return k.Exists(ctx, key)' instead of 'if k.Exists(ctx, key) { return true }; return false' (S1008)
x/dispensation/keeper/userClaim.go:33:2: should use 'return k.Exists(ctx, key)' instead of 'if k.Exists(ctx, key) { return true }; return false' (S1008)
x/dispensation/types/records.go:103:2: should use 'return ok' instead of 'if !ok { return false }; return true' (S1008)
x/dispensation/types/records.go:191:2: should use 'return err == nil' instead of 'if err != nil { return false }; return true' (S1008)
x/dispensation/types/records.go:223:2: should use 'return ok' instead of 'if !ok { return false }; return true' (S1008)
x/faucet/types/params.go:38:9: unnecessary use of fmt.Sprintf (S1039)
x/oracle/keeper/keeper.go:54:6: should use !bytes.Equal(key, types.AdminAccountPrefix) instead (S1004)
x/oracle/keeper/keeper.go:55:4: should use !bytes.Equal(key, types.WhiteListValidatorPrefix) instead (S1004)
[redacted] Downloads %
```

Errcheck - Security Analysis Output Sample:

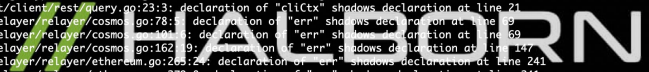
```
halborn@zion:~/Downloads/sifnode$ ../../go/bin/errcheck ./...
cmd/ebrelayer/relayer/cosmos.go:86:19:  defer client.Stop() //nolint:errcheck
cmd/ebrelayer/relayer/cosmos.go:318:19: defer client.Stop() //nolint:errcheck
cmd/ebrelayer/relayer/ethereum.go:313:19:       defer client.Stop() //nolint:errcheck
cmd/ebrelayer/relayer/listMissedEvent.go:93:19: defer client.Stop() //nolint:errcheck
cmd/ebrelayer/txs/parser_test.go:57:11: os.Setenv(EthereumPrivateKey, TestPrivHex)
cmd/ebrelayer/txs/signature_test.go:42:11:      os.Setenv(EthereumPrivateKey, TestPrivHex)
tools/sifgen/genesis/genesis.go:119:27: defer response.Body.Close()
tools/sifgen/node/node.go:281:27:       defer response.Body.Close()
[redacted]
```

Unconvert - Security Analysis Output Sample:

```
unconvert ./...
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/genesis.pb.go:241:26: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/params.pb.go:115:41: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/params.pb.go:140:28: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/params.pb.go:149:25: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/querier.pb.go:1678:26: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/tx.pb.go:1354:21: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/clp/types/types.pb.go:517:24: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/dispensation/types/query.pb.go:1072:24: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/dispensation/types/tx.pb.go:817:21: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/dispensation/types/types.pb.go:1056:24: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/ethbridge/types/query.pb.go:489:24: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/ethbridge/types/tx.pb.go:1564:21: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/ethbridge/types/types.pb.go:579:24: unnecessary conversion
/Users/tag/src/github.com/Sifchain/sifnode/x/oracle/types/types.pb.go:639:24: unnecessary conversion
```

Shadow - Security Analysis Output Sample:

```
/home/destek/Downloads/sifnode/x/clp/client/rest/query.go:40:3: declaration of "cliCtx" shadows declaration at line 38
/home/destek/Downloads/sifnode/x/clp/client/rest/query.go:68:3: declaration of "cliCtx" shadows declaration at line 66
/home/destek/Downloads/sifnode/x/clp/client/rest/query.go:100:3: declaration of "cliCtx" shadows declaration at line 98
/home/destek/Downloads/sifnode/x/clp/client/rest/query.go:120:3: declaration of "cliCtx" shadows declaration at line 118
/home/destek/Downloads/sifnode/x/clp/client/rest/query.go:151:3: declaration of "cliCtx" shadows declaration at line 149
/home/destek/Downloads/sifnode/x/clp/handler.go:71:3: declaration of "err" shadows declaration at line 37
/home/destek/Downloads/sifnode/x/clp/handler.go:224:34: declaration of "err" shadows declaration at line 185
/home/destek/Downloads/sifnode/x/clp/handler.go:245:34: declaration of "err" shadows declaration at line 185
/home/destek/Downloads/sifnode/x/clp/handler.go:326:34: declaration of "err" shadows declaration at line 304
/home/destek/Downloads/sifnode/x/dispensation/client/rest/query.go:43:3: declaration of "cliCtx" shadows declaration at line 41
/home/destek/Downloads/sifnode/x/dispensation/client/rest/query.go:62:3: declaration of "cliCtx" shadows declaration at line 60
/home/destek/Downloads/sifnode/x/dispensation/client/rest/query.go:93:3: declaration of "cliCtx" shadows declaration at line 91
/home/destek/Downloads/sifnode/x/dispensation/client/rest/query.go:124:3: declaration of "cliCtx" shadows declaration at line 122
/home/destek/Downloads/sifnode/x/dispensation/keeper/executors.go:43:4: declaration of "err" shadows declaration at line 41
/home/destek/Downloads/sifnode/x/ethbridge/types/test_common.go:109:2: declaration of "MsgRescueCeth" shadows declaration at line 277
/home/destek/Downloads/sifnode/x/faucet/client/rest/query.go:23:3: declaration of "cliCtx" shadows declaration at line 21
/home/destek/Downloads/sifnode/cmd/ebrelayer/relayer/cosmos.go:78:[illegible]: declaration of "err" shadows declaration at line 69
/home/destek/Downloads/sifnode/cmd/ebrelayer/relayer/cosmos.go:[illegible]: declaration of "err" shadows declaration at line 69
/home/destek/Downloads/sifnode/cmd/ebrelayer/relayer/cosmos.go:162:9: declaration of "err" shadows declaration at line 147
/home/destek/Downloads/sifnode/cmd/ebrelayer/relayer/ethereum.go:203:24: declaration of "err" shadows declaration at line 241
/home/destek/Downloads/sifnode/cmd/ebrelayer/relayer/ethereum.go:279:8: declaration of "err" shadows declaration at line 241
/home/destek/Downloads/sifnode/cmd/ebrelayer/main.go:120:6: declaration of "err" shadows declaration at line 115
/home/destek/Downloads/sifnode/cmd/ebrelayer/main.go:140:6: declaration of "err" shadows declaration at line 115
/home/destek/Downloads/sifnode/cmd/ebrelayer/main.go:176:6: declaration of "err" shadows declaration at line 115
/home/destek/Downloads/sifnode/cmd/ebrelayer/replay.go:38:6: declaration of "err" shadows declaration at line 24
/home/destek/Downloads/sifnode/cmd/ebrelayer/replay.go:113:6: declaration of "err" shadows declaration at line 105
/home/destek/Downloads/sifnode/tools/sifgen/genesis/genesis.go:127:5: declaration of "err" shadows declaration at line 109
/home/destek/Downloads/sifnode/tools/sifgen/network/network.go:41:5: declaration of "err" shadows declaration at line 29
/home/destek/Downloads/sifnode/tools/sifgen/network/network.go:343:9: declaration of "err" shadows declaration at line 338
/home/destek/Downloads/sifnode/tools/sifgen/node/node.go:110:5: declaration of "err" shadows declaration at line 85
/home/destek/Downloads/sifnode/tools/sifgen/node/node.go:162:6: declaration of "err" shadows declaration at line 149
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:58:9: declaration of "err" shadows declaration at line 54
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:93:25: declaration of "err" shadows declaration at line 54
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:112:7: declaration of "err" shadows declaration at line 54
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:181:9: declaration of "err" shadows declaration at line 177
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:268:25: declaration of "err" shadows declaration at line 251
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:287:7: declaration of "err" shadows declaration at line 251
/home/destek/Downloads/sifnode/cmd/sifnoded/genaccounts.go:412:9: declaration of "err" shadows declaration at line 408
/home/destek/Downloads/sifnode/x/dispensation/keeper/querier_test.go:20:3: declaration of "name" shadows declaration at line 17
```



According to the test results, some of the findings found by these tools were considered as false positives while some of these findings were real security concerns. All relevant findings were reviewed by the auditors and relevant findings addressed on the report as security concerns.

THANK YOU FOR CHOOSING

// HALBORN