// HALBORN

A41 - SuperNova Bot

Golang Security Audit

Prepared by: **Halborn**

Date of Engagement: **November 14th, 2022 - November 23rd, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	11/15/2022	Gokberk Gulgun
0.2	Document Updates	11/21/2022	Gokberk Gulgun
0.3	Draft Review	11/23/2022	Gabi Urrutia
1.0	Remediation Plan	11/25/2022	Gokberk Gulgun
1.1	Remediation Plan Review	11/28/2022	Gabi Urrutia

CONTACTS

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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

A41 Team engaged Halborn to conduct a security audit on their **SuperNova Bot**, beginning on November 14th, 2022 and ending on November 23rd, 2022 . The security assessment was scoped to the code base provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided nearly five weeks for the engagement and assigned two full-time security engineers to audit the security of the **modules**. 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 NOVA Bot functionalities as intended.
- Identify potential security issues with the A41 Team.

In summary, Halborn identified few security risks that were mostly addressed by the A41 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 module. 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 module 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.

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

1.4 SCOPE

IN-SCOPE:

The security assessment was scoped to Carina-labs/HAL9000 repository.

TREE

FIX COMMIT IDs :

- a6a44a376
- 5165f8b7e
- 41700f77
- 273c89b
- 138541f
- 8cc34a0d
- 4ae5fa
- 576ac33

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	2	1	5	4

LIKELIHOOD

IMPACT

		(HAL-01) (HAL-02)		
(HAL-04) (HAL-05) (HAL-08)		(HAL-03)		
	(HAL-06) (HAL-07)			
(HAL-09) (HAL-10) (HAL-11) (HAL-12)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - BOT INCONSISTENTLY CAN FAIL DUE TO MISSING CHECK	High	SOLVED - 11/25/2022
HAL-02 - LACK OF ERROR CHECKS CAN LEADS TO UNEXCEPTED FAILURES ON THE BOT	High	SOLVED - 11/25/2022
HAL-03 - ReadHeaderTimeout IS NOT SET IN THE HTTP SERVE	Medium	SOLVED - 11/25/2022
HAL-04 - IBC TIMEOUT IS NOT COMPATIBLE WITH THE CHAIN	Low	SOLVED - 11/25/2022
HAL-05 - TEST DOCKERFILE IS RUNNING AS ROOT	Low	SOLVED - 11/25/2022
HAL-06 - USE WEAK RANDOM NUMBER GENERATOR	Low	SOLVED - 11/25/2022
HAL-07 - SECRET YAML SHOULD BE ADDED INTO GIT IGNORE	Low	SOLVED - 11/25/2022
HAL-08 - HARDCODED USE OF INSECURE GRPC TRANSPORT	Low	SOLVED - 11/25/2022
HAL-09 - LACK OF EXTENSIVE TEST COVERAGE	Informational	ACKNOWLEDGED
HAL-10 - OPEN TODO IN CODEBASE	Informational	SOLVED - 11/25/2022
HAL-11 - SPELLING MISTAKES IN THE CODEBASE	Informational	SOLVED - 11/25/2022
HAL-12 - VULNERABLE THIRD PARTY PACKAGES	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) BOT INCONSISTENTLY CAN FAIL DUE TO MISSING CHECK - HIGH

Description:

On the **SuperNova Bot**, several actions can be completed through the binary. In the Supernova protocol, The oracle module manages the status of the zones associated. The status includes the amount of coins delegated to the Zone's **Validator**, **block height**, **proof**. With UpdateChainState message, oracle module updates the status of the zones stored in Oracle with a new status. In the **SuperNova** Bot, Transaction error is not defined for UpdateChainState message by the **BOT**. The bot can fail silently due to missing control.

Code Location:

/logic/router.go#L11

Listing 1

```
1 func RouteBotAction(botType string, b *basetypes.Bot, cni *config.
↳ ChainNetInfo, hci *config.HostChainInfo) {
2     initialBanner(botType)
3     switch botType {
4     case config.ActOracle:
5         cq := query.NewCosmosQueryClient(hci.GrpcAddr)
6         defer utils.CloseGrpc(cq.ClientConn)
7         UpdateChainState(cq, b, hci)
8     case config.ActStake:
9         nq := novaq.NewNovaQueryClient(cni.GRPC.Host)
10        defer utils.CloseGrpc(nq.ClientConn)
11        IcaStake(nq, b, hci)
12    case config.ActAutoStake:
13        cq := query.NewCosmosQueryClient(hci.GrpcAddr)
14        nq := novaq.NewNovaQueryClient(cni.GRPC.Host)
15        defer utils.CloseGrpc(cq.ClientConn)
16        defer utils.CloseGrpc(nq.ClientConn)
17        IcaAutoStake(cq, nq, b, hci)
18    case config.ActWithdraw:
```

```
19          cq := query.NewCosmosQueryClient(hci.GrpcAddr)
20          nq := novaq.NewNovaQueryClient(cni.GRPC.Host)
21          defer utils.CloseGrpc(cq.ClientConn)
22          defer utils.CloseGrpc(nq.ClientConn)
23          UndelegateAndWithdraw(cq, nq, b, hci)
24      default:
25          panic("This type cannot handle at this action router")
26
```

Risk Level:

Likelihood - 3
Impact - 5

Proof Of Concept:

Listing 2

```
1  func TestMsgUpdateChainStateValidation(t *testing.T) {
2      addr := sdk.AccAddress([]byte("addr_____________"))
3
4      msg := MsgUpdateChainState{
5          Coin:        sdk.NewCoin("atom", sdk.NewInt(1000)),
6          Operator:    addr.String(),
7          BlockHeight: 10,
8          AppHash:     []byte("apphash"),
9          ZoneId:      "cosmos",
10     }
11     err := msg.ValidateBasic()
12     require.NoError(t, err)
13
14     // check invalid address
15     msg.Operator = "invalid"
16     err = msg.ValidateBasic()
17     require.Error(t, err)
18 }
```

/client/base/bot.go#L88

Listing 3

```
func handleTxErr(f *os.File, e error) TxErr {
    if e != nil {
        if strings.Contains(e.Error(), "account sequence mismatch") ) {
            utils.LogErrWithFd(f, e, "  ", ut.KEEP)
            return SEQMISMATCH
        } else if strings.Contains(e.Error(), "cannot change state ") {
            utils.LogErrWithFd(f, e, "  There is no asset to delegate on this host zone   go to next batch\n", ut.KEEP)
            return NEXT
        } else if strings.Contains(e.Error(), "invalid coins") {
            utils.LogErrWithFd(f, e, "  There is no reward to autostake on this host zone   go to next batch\n", ut.KEEP)
            return NEXT
        } else if strings.Contains(e.Error(), "no coins to undelegate") {
            utils.LogErrWithFd(f, e, "  There is no asset to undelegate on this host zone   go to next batch\n", ut.KEEP)
            return NEXT
        } else if strings.Contains(e.Error(), "cannot withdraw funds") {
            utils.LogErrWithFd(f, e, "  There is no asset to withdraw on this host zone   go to next batch\n", ut.KEEP)
            return NEXT
        } else if strings.Contains(e.Error(), "current block height must be higher than the previous block height") {
            utils.LogErrWithFd(f, e, "  oracle info was outdated due to the oracle bot's update. It will regenerate tx\n", ut.KEEP)
            return REPEAT
        }

        utils.LogErrWithFd(f, e, "  something went wrong while generate tx", ut.KEEP)
        return NORMAL
    }
    return NONE
}
```

Recommendation:

Make sure all errors are handled in transactions.

Listing 4

```
var (
    ErrNoSupportChain      = sdkerrors.Register(ModuleName, 0, "this chain is not supported")
    ErrInvalidOperator     = sdkerrors.Register(ModuleName, 1, "invalid operator address")
    ErrUnknown             = sdkerrors.Register(ModuleName, 2, "unknown error")
    ErrNotFoundZoneInfo    = sdkerrors.Register(ModuleName, 3, "not found zone info")
    ErrInvalidKeyManager   = sdkerrors.Register(ModuleName, 4, "invalid key manager address")
    ErrNegativeBlockHeight = sdkerrors.Register(ModuleName, 5, "blockHeight must be positive")
    ErrInvalidBlockHeight  = sdkerrors.Register(ModuleName, 6, "current block height must be higher than the previous block height.")
)
```

Remediation Plan:

SOLVED: The A41 team solved the issue in commit a6a44a376 by adding the error messages.

3.2 (HAL-02) LACK OF ERROR CHECKS CAN LEADS TO UNEXCEPTED FAILURES ON THE BOT - HIGH

Description:

In the **SuperNova Bot**, the following actions can be completed through the bot.

Listing 5

```
oracle : Update host's base token price every 15 minutes.

withdraw : Undelegate and withdraw token from host account to nova. The interval depends on the rules of the host chain.

stake : Delegate the tokens sent by the user to the host chain via IBC to the a4x validator through the controller account every 10 mintues.

restake : Automatically re-stake the host account's rewards through IBC. The amount to be re-deposited is inquired from the distribution module of the host chain every 6 hours.
```

There are some instances where error handling has not been implemented for the functions that might return an error. Without the error handling, the operations can silently fail and can’t continue the workflow.

Code Location:

/logic/tx.go#L82-L157

Listing 6

```
func UpdateChainState(cq *query.CosmosQueryClient, b *novatypes.Bot, host *config.HostChainInfo) {
    i := 0
    intv := time.Duration(b.Interval)
```

```
    for {
        _ = mustExecTx(b, host, msgs)
    }
}

func IcaAutoStake(cq *query.CosmosQueryClient, nq *novaq.
↳ NovaQueryClient, b *novatypes.Bot, host *config.HostChainInfo) {
...
        _ = mustExecTx(b, host, msgs, IBCConfirm{nq, config.
↳ ActAutoStake, targetSeq})
...
    }

func IcaAutoStake(cq *query.CosmosQueryClient, nq *novaq.
↳ NovaQueryClient, b *novatypes.Bot, host *config.HostChainInfo) {
...
        _ = mustExecTx(b, host, msgs, IBCConfirm{nq, config.
↳ ActAutoStake, targetSeq})
...
    }

func IcaStake(nq *novaq.NovaQueryClient, b *novatypes.Bot, host *
↳ config.HostChainInfo) {
...
        _ = mustExecTx(b, host, msgs, IBCConfirm{nq, config.
↳ ActStake, targetSeq})
...
    }

func UndelegateAndWithdraw(cq *query.CosmosQueryClient, nq *novaq.
↳ NovaQueryClient, b *novatypes.Bot, host *config.HostChainInfo) {
...
            _ = mustExecTx(b, host, msgs, IBCConfirm{nq, config.
↳ ActWithdraw, wdSeq})
...
        }()
```

Risk Level:

Likelihood - 3
Impact - 5

Proof Of Concept:

Listing 7

```
func TestMsgUpdateChainStateValidation(t *testing.T) {
    addr := sdk.AccAddress([]byte("addr_________________"))

    msg := MsgUpdateChainState{
        Coin:        sdk.NewCoin("atom", sdk.NewInt(1000)),
        Operator:    addr.String(),
        BlockHeight: 10,
        AppHash:     []byte("apphash"),
        ZoneId:      "cosmos",
    }
    err := msg.ValidateBasic()
    require.NoError(t, err)

    // check invalid address
    msg.Operator = "invalid"
    err = msg.ValidateBasic()
    require.Error(t, err)
}
```

Recommendation:

It is recommended to implement proper error checking to avoid unexpected crashes.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit a6a44a376 by adding the error checks.

3.3 (HAL-03) ReadHeaderTimeout IS NOT SET IN THE HTTP SERVE - MEDIUM

Description:

Slowloris is a type of denial of service (DoS) attack tool which allows a single machine to take down another machine's web server with minimal bandwidth and side effects on unrelated services and ports. **Slowloris** tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. Configuring **ReadHeaderTimeout** would protect directly against this attack by closing the connection once the deadline is reached. By default, Go does not define any value meaning there is no timeout.

Code Location:

/api/server.go#L54

Listing 8

```
1 func (s Server) On(addr string) {
2
3     http.Handle("/metrics", promhttp.Handler())
4     http.Handle("/check/", NewChkHandler())
5     err := http.ListenAndServe(addr, nil)
6     utils.CheckErr(err, "cannot open http server", ut.EXIT)
7 }
8
```

Risk Level:

Likelihood - 3
Impact - 3

Proof Of Concept:

```
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
dial tcp :80: socket: too many open files
```

Recommendation:

It is recommended to define **ReadHeaderTimeout** in the API.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 5165f8b7e by adding **ReadHeaderTimeout** in the API.

3.4 (HAL-04) IBC TIMEOUT IS NOT COMPATIBLE WITH THE CHAIN - LOW

Description:

In the codebase, it is mentioned as **This value must be set higher than the IBC timeout**, but the value is incompatible with SuperNova Chain. The incompatibility can lead to error on the transaction generation.

Code Location:

/logic/params.go#L9

Listing 9

```
 1 package logic
 2
 3 import (
 4     "time"
 5 )
 6
 7 const (
 8     ReQueryDelay = time.Second * 1
 9     IBCDelay     = time.Second * 30 // This value must be set
↳ higher than the ibc timeout.
10 )
11
12 const (
13     QueryErrPrefix = "[QUERY ERROR] : "
14 )
15
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Ensure that **IBCDelay** is compatible with SuperNova Chain.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 41700f77 by adding **IBC Timeout** as a flag.

3.5 (HAL-05) TEST DOCKER IMAGE RUNNING AS ROOT - LOW

Description:

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

Code Location:

Dockerfile

Listing 10

```
1  FROM golang:1.19-alpine AS builder
2
3  FROM builder AS builder-amd64
4  ARG arch=x86_64
5
6  FROM builder AS builder-arm64
7  ARG arch=aarch64
8
9  FROM builder-$TARGETARCH AS release
10 RUN set -eux; apk add --no-cache ca-certificates build-base;
11 RUN apk add git
12 ARG GHTOKEN
13 RUN git config --global url."https://$GHTOKEN@github.com/".
↳ insteadOf "https://github.com/" && go env -w GOPRIVATE=github.com/
↳ Carina-labs
14 WORKDIR /workspace
15 COPY . .
16 ADD https://github.com/CosmWasm/wasmvm/releases/download/v1.1.1/
↳ libwasmvm_muslc.aarch64.a /lib/libwasmvm_muslc.aarch64.a
17 ADD https://github.com/CosmWasm/wasmvm/releases/download/v1.1.1/
↳ libwasmvm_muslc.x86_64.a /lib/libwasmvm_muslc.x86_64.a
18 RUN sha256sum /lib/libwasmvm_muslc.aarch64.a | grep 9
↳ ecb037336bd56076573dc18c26631a9d2099a7f2b40dc04b6cae31ffb4c8f9a
```

```
19 RUN sha256sum /lib/libwasmvm_muslc.x86_64.a | grep 6
↳ e4de7ba9bad4ae9679c7f9ecf7e283dd0160e71567c6a7be6ae47c81ebe7f32
20 RUN cp /lib/libwasmvm_muslc.${arch}.a /lib/libwasmvm_muslc.a
21 RUN LINK_STATICALLY=true make build
22
23 FROM alpine:3.16
24 RUN apk add --update --no-cache  ca-certificates libstdc++ yq
25 ENV TARGET=hal
26 ENV PATH="${PATH}:/workspace"
27 WORKDIR /workspace
28 COPY --from=release /workspace/build/$TARGET ./$TARGET
29 # comment out below if you need config dynamic linking
30 COPY .chaininfo.yaml .secret.yaml ./config/
31 CMD ["hal", "--help"]
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

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

Listing 11: Reference

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

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 273c89b by changing user on the **Dockerfile**.

3.6 (HAL-06) USE OF WEAK RANDOM GENERATOR - LOW

Description:

When a non-cryptographic PRNG is used in a cryptographic context, it can expose the cryptography to certain types of attacks. Often a pseudo-random number generator (PRNG) is not designed for cryptography. Sometimes a poor source of randomness is enough or preferable for algorithms that use random numbers. Weak generators generally take less processing power and/or do not use the precious, finite, entropy sources on a system. While such PRNGs might have very useful features, these same features could be used to break the cryptography.

Code Location:

/rpc/types/ws.go#L8-L9

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

Consider replacing math/rand with crypto/rand.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 138541f by using crypto/rand.

3.7 (HAL-07) SECRET YAML SHOULD BE ADDED INTO GIT IGNORE - LOW

Description:

.secret.yml is used to pipe keyring password. A gitignore file specifies intentionally untracked files that Git should ignore. All sensitive information should be added into the gitignore to prevent commit into repository.

Code Location:

/.secret.yaml

Listing 12: Reference

```
1 pw: masked
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

Consider removing .secret.yml and adding it to .gitignore.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 8cc34a0d by adding .secret.yml into .gitignore.

3.8 (HAL-08) HARDCODED USE OF INSECURE GRPC TRANSPORT - LOW

Description:

During the code review, it was noted that **gRPC** client uses a hardcoded `WithInsecure()` transport setting when dialing a remote. This could allow man-in-the-middle attacks between the gRPC client and server.

Code Location:

/client/nova/query/querier.go#L30

Listing 13

```
1 func NewNovaQueryClient(grpcAddr string) *NovaQueryClient {
2     conn, err := grpc.Dial(
3         grpcAddr,
4         grpc.WithInsecure(),
5     )
6     utils.CheckErr(err, "cannot create gRPC connection", 0)
7     return &NovaQueryClient{conn}
8 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendations:

In the short term, add documentation that explains to end users the simplest mechanism to secure gRPC. In the long term, consider adding a configuration option that allows you to select gRPC transport as secure or non-secure, where secure transport is the default.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 576ac33 by adding secure option.

3.9 (HAL-09) LACK OF EXTENSIVE TEST COVERAGE - INFORMATIONAL

Description:

Adequate test coverage and regular reporting is an essential process to ensure the codebase works as intended. Insufficient code coverage can lead to unexpected issues and regressions due to changes in module implementations.

Code Location:

SuperNova Bot

Recommendation:

Make sure the coverage report produced via **go test -cover** covers all functions.

Remediation Plan:

ACKNOWLEDGED: The A41 team acknowledged this issue.

3.10 (HAL-10) OPEN TODO IN CODEBASE - INFORMATIONAL

Description:

Open To-dos can point to architecture or programming issues that still need to be resolved. Often these kinds of comments indicate areas of complexity or confusion for developers. This provides value and insight to an attacker who aims to cause damage to the protocol.

Code Location:

Listing 14: Open Todos

```
1 ./logic/tx.go:43:                // TODO: Need to implement a
↳ system that can resolve this issue more effectively than simply
↳ showing the log
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Consider resolving any pending tasks before deploying the code to a production context. Use an independent issue tracker or other project management software to keep track of development tasks.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 576ac33 by resolving to-dos.

3.11 (HAL-11) SPELLING MISTAKES IN THE CODEBASE - INFORMATIONAL

Description:

Spelling mistakes were identified within the codebase.

Code Location:

/base/tx_test.go#L47

Listing 15: Reference

```
            assert.Errorf(s.T(), err, "cannot covert target to
↳ AccAddress")
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended that all filenames and word usage within the code be spelled correctly, as this will avoid confusion during development. Proper spelling can also help convey a sense of professionalism to the various project stakeholders.

Remediation Plan:

SOLVED: The A41 team solved the issue in commit 4ae5fa by fixing the typo.

3.12 (HAL-12) VULNERABLE THIRD PARTY PACKAGES - INFORMATIONAL

Description:

During the audit, Halborn identified installed 3rd party packages that contain known security vulnerabilities.

Packages:

ID	Package	Rating
CVE-2022-32149	text@v0.3.7	HIGH
sonatype-2021-0598	tendermint@v0.34.12	MEDIUM
CVE-2022-44797	btcd@v0.22.1	HIGH
CVE-2022-39389	btcd@v0.22.1	MEDIUM

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to keep all installed third-party packages up to date and apply all security fixes.

Remediation Plan:

ACKNOWLEDGED: The A41 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 16: 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 17

```
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
```

```
 4         hazard. If an attacker can control a process running as
↳ root, they may have control over the
 5         container. Ensure that the last USER in a Dockerfile is a
↳ USER other than 'root'.
 6         Details: https://sg.run/Gbvn
 7
 8          31 CMD ["hal", "--help"]
 9
10
11   api/controller.go
12      go.lang.security.audit.xss.no-io-writestring-to-
↳ responsewriter.no-io-writestring-to-
13      responsewriter
14         Detected 'io.WriteString()' writing directly to 'http.
↳ ResponseWriter'. This bypasses HTML
15         escaping that prevents cross-site scripting
↳ vulnerabilities. Instead, use the
16         'html/template' package to render data to users.
17         Details: https://sg.run/gLwn
18
19          22 _, err := io.WriteString(w, BotStatus.LastCommit.
↳ String())
20
21
22   client/base/query/querier.go
23      go.grpc.security.grpc-client-insecure-connection.grpc-client-
↳ insecure-connection
24         Found an insecure gRPC connection using 'grpc.WithInsecure
↳ ()'. This creates a connection
25         without encryption to a gRPC server. A malicious attacker
↳ could tamper with the gRPC
26         message, which could compromise the machine. Instead,
↳ establish a secure connection with an
27         SSL certificate using the 'grpc.WithTransportCredentials()
↳ ' function. You can create a
28         create credentials using a 'tls.Config{}' struct with '
↳ credentials.NewTLS()'. The final fix
29         looks like this: 'grpc.WithTransportCredentials(
↳ credentials.NewTLS(<config>))'.
30         Details: https://sg.run/J9yZ
31
32           Autofix  s/(.*)WithInsecure\(.*?\)/\1
↳ WithTransportCredentials(credentials.NewTLS(<your_tls_config_here
↳ >))/g
```

```
33          32 conn, err := grpc.Dial(
34          33     grpcAddr,
35          34     grpc.WithInsecure(),
36          35 )
37
38
39    client/nova/query/querier.go
40       go.grpc.security.grpc-client-insecure-connection.grpc-client-
↳ insecure-connection
41          Found an insecure gRPC connection using 'grpc.WithInsecure
↳ ()'. This creates a connection
42          without encryption to a gRPC server. A malicious attacker
↳ could tamper with the gRPC
43          message, which could compromise the machine. Instead,
↳ establish a secure connection with an
44          SSL certificate using the 'grpc.WithTransportCredentials()
↳ ' function. You can create a
45          create credentials using a 'tls.Config{}' struct with '
↳ credentials.NewTLS()'. The final fix
46          looks like this: 'grpc.WithTransportCredentials(
↳ credentials.NewTLS(<config>))'.
47          Details: https://sg.run/J9yZ
48
49            Autofix  s/(.*)WithInsecure\(.*?\)/\1
↳ WithTransportCredentials(credentials.NewTLS(<your_tls_config_here
↳ >))/g
50          28 conn, err := grpc.Dial(
51          29     grpcAddr,
52          30     grpc.WithInsecure(),
53          31 )
54
```

Gosec - Security Analysis Output Sample:

Listing 18

```
1
2 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/rpc/types/ws.go:272] -
↳ G404 (CWE-338): Use of weak random number generator (math/rand
↳ instead of crypto/rand) (Confidence: MEDIUM, Severity: HIGH)
3     271:         // nolint:gosec // G404: Use of weak random number
↳  generator
4   > 272:         jitter := time.Duration(mrand.Float64() * float64(
↳ time.Second)) // 1s == (1e9 ns)
5     273:         backoffDuration := jitter + ((1 << attempt) * time
↳ .Second)
6
7
8
9 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/api/server.go:54] - G114
↳ (CWE-676): Use of net/http serve function that has no support for
↳ setting timeouts (Confidence: HIGH, Severity: MEDIUM)
10    53:      http.Handle("/check/", NewChkHandler())
11  > 54:      err := http.ListenAndServe(addr, nil)
12    55:      utils.CheckErr(err, "cannot open http server", ut.EXIT
↳ )
13
14
15
16 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:64] - G304
↳ (CWE-22): Potential file inclusion via variable (Confidence: HIGH,
↳  Severity: MEDIUM)
17    63:          //   fmt.Fprintf(os.stderr)
18  > 64:          fdErrExt, err = os.OpenFile(path.Join(logDir,
↳ errRedirectLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
19    65:          utils.CheckErr(err, "cannot open otherErr", 0)
20
21
22
23 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:60] - G304
↳ (CWE-22): Potential file inclusion via variable (Confidence: HIGH,
↳  Severity: MEDIUM)
24    59:          //
```

```
25   > 60:          fdErr, err = os.OpenFile(path.Join(logDir,
↳ errLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
26     61:          utils.CheckErr(err, "cannot open novaerr", 0)
27
28
29
30 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:56] - G304
↳ (CWE-22): Potential file inclusion via variable (Confidence: HIGH,
↳  Severity: MEDIUM)
31     55:      if !isDisp {
32   > 56:          fdLog, err = os.OpenFile(path.Join(logDir,
↳ stdLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
33     57:          utils.CheckErr(err, "cannot open logfp", 0)
34
35
36
37 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:64] - G302
↳ (CWE-276): Expect file permissions to be 0600 or less (Confidence:
↳  HIGH, Severity: MEDIUM)
38     63:          //   fmt.Fprintf(os.stderr)
39   > 64:          fdErrExt, err = os.OpenFile(path.Join(logDir,
↳ errRedirectLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
40     65:          utils.CheckErr(err, "cannot open otherErr", 0)
41
42
43
44 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:60] - G302
↳ (CWE-276): Expect file permissions to be 0600 or less (Confidence:
↳  HIGH, Severity: MEDIUM)
45     59:          //
46   > 60:          fdErr, err = os.OpenFile(path.Join(logDir,
↳ errLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
47     61:          utils.CheckErr(err, "cannot open novaerr", 0)
48
49
50
51 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/config/bot.go:56] - G302
↳ (CWE-276): Expect file permissions to be 0600 or less (Confidence:
↳  HIGH, Severity: MEDIUM)
52     55:      if !isDisp {
```

```
53   > 56:          fdLog, err = os.OpenFile(path.Join(logDir,
↳ stdLogName), os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0644)
54     57:          utils.CheckErr(err, "cannot open logfp", 0)
55
56
57
58 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/rpc/types/ws.go:429] -
↳ G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW
↳ )
59     428:      defer func() {
60   > 429:          c.conn.Close()
61     430:          // err != nil {
62
63
64
65 [/Users/Halborn/Downloads/Projects/HAL9000-
↳ adde694a85250e59f64511e7baade825e578c825/rpc/types/ws.go:373] -
↳ G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW
↳ )
66     372:          ticker.Stop()
67   > 373:          c.conn.Close()
68     374:          // err != nil {
69
70
```

Staticcheck - Security Analysis Output Sample:

Listing 19

```
1 No finding
```

THANK YOU FOR CHOOSING

// HALBORN