HALBORN

StaderLabs - Permissionless Stader Node

Golang Security Audit

Prepared by: **Halborn**

Date of Engagement: **June 1st, 2023 - June 27th, 2023**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY 7

CONTACTS 8

1 EXECUTIVE OVERVIEW 9

1.1 INTRODUCTION 10

1.2 AUDIT SUMMARY 10

1.3 SCOPE 11

1.4 TEST APPROACH & METHODOLOGY 12

2 RISK METHODOLOGY 13

2.1 EXPLOITABILITY 14

2.2 IMPACT 15

2.3 SEVERITY COEFFICIENT 17

3 ASSESSMENT SUMMARY & FINDINGS OVERVIEW 19

4 FINDINGS & TECH DETAILS 21

4.1 (HAL-01) POTENTIAL INTEGER OVERFLOW FOR CYCLES IN CLAIM REWARDS - LOW(3.9) 23

Description 23

Code Location 23

BVSS 24

Recommendation 24

Remediation Plan 25

4.2 (HAL-02) POTENTIAL INTEGER OVERFLOW FOR LARGE NONCES AND BLOCK NUMBERS - LOW(3.9) 26

Description 26

Code Location 26

BVSS 29

Recommendation 29

Remediation Plan 29

4.3 (HAL-03) USE OF ARCHITECTURE-SPECIFIC INTEGER TYPE COULD CAUSE PROBLEMS ON 32 BIT SYSTEMS - LOW(3.9) 30

Description 30

Code Location 30

BVSS 31

Recommendation 31

Remediation Plan 31

4.4 (HAL-04) PRECISION LOSS DUE TO DIVISION OPERATION OCCURRING BEFORE MULTIPLICATION - LOW(3.1) 32

Description 32

Code Location 32

BVSS 33

Recommendation 33

Remediation Plan 33

4.5 (HAL-05) FILES AND FOLDERS CREATED BY THE VALIDATOR DO NOT FOLLOW THE PRINCIPLE OF LEAST PRIVILEGE - INFORMATIONAL(1.5) 34

Description 34

Code Location 34

BVSS 34

Recommendation 35

Remediation Plan 35

4.6 (HAL-06) USE OF UNSUPPORTED GO VERSION - INFORMATIONAL(0.0) 36

Description 36

Code Location 36

BVSS 36

Recommendation 36
Remediation Plan 36
4.7 (HAL-07) USE OF VULNERABLE DEPENDENCIES - INFORMATIONAL(0.0) 37
Description 37
Code Location 37
BVSS 37
Recommendation 37
Remediation Plan 38
4.8 (HAL-08) ERROR MESSAGE REPORTS INCORRECT CRYPTOGRAPHIC ALGORITHM - INFORMATIONAL(0.0) 39
Description 39
Code Location 39
BVSS 40
Recommendation 40
Remediation Plan 40
4.9 (HAL-09) FUNCTION DECRYPTUSINGPUBLICKEY USES PRIVATEKEY INSTEAD OF PUBLICKEY AS A PARAMETER - INFORMATIONAL(0.0) 41
Description 41
Code Location 41
BVSS 41
Recommendation 42
Remediation Plan 42
4.10 (HAL-10) FUNCTION DECRYPTUSINGPUBLICKEY IS UNUSED - INFORMATIONAL(0.0) 43
Description 43
Code Location 43
BVSS 43

Recommendation 43
Remediation Plan 44
4.11 (HAL-11) TODOS IN CODEBASE - INFORMATIONAL(0.0) 45
Description 45
Code Location 45
BVSS 45
Recommendation 45
Remediation Plan 46
4.12 (HAL-12) USE OF FORMATTING SYMBOL IN PRINTLN - INFORMATIONAL(0.0) 47
Description 47
Code Location 47
BVSS 47
Recommendation 47
Remediation Plan 47
4.13 (HAL-13) VOLUNTARY EXIT MESSAGES CAN EXPIRE - INFORMATIONAL(0.0) 48
Description 48
Code Location 48
BVSS 49
Recommendation 49
Remediation Plan 49
4.14 (HAL-14) USE TLS IN LISTENER INSTEAD OF PLAIN HTTP - INFORMATIONAL(0.0) 50
Description 50
Code Location 50
BVSS 50

Recommendation 51
Remediation Plan 51
4.15 (HAL-15) ITERATION OVER A MAP MAY CAUSE ISSUES WITH VALIDATOR STORAGE - INFORMATIONAL(0.0) 52
Description 52
Code Location 52
BVSS 56
Recommendation 56
Remediation Plan 56
4.16 (HAL-16) FLOATING POINT ARITHMETIC IS NON-DETERMINISTIC - INFORMATIONAL(0.0) 57
Description 57
Code Location 57
BVSS 58
Recommendation 58
Remediation Plan 58
4.17 (HAL-17) UNHANDLED ERRORS - INFORMATIONAL(0.0) 59
Description 59
Code Location 59
BVSS 61
Recommendation 61
Remediation Plan 61
5 AUTOMATED TESTING 62
Description 63
Semgrep 63
Gosec 67
CodeQL 71

Govulncheck 72

Nancy 72

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	06/16/2023	Alejandro Taibo
0.2	Document Updates	06/26/2023	John Saigle
0.3	Draft Version	06/27/2023	Alejandro Taibo
0.4	Draft Review	06/27/2023	Gokberk Gulgun
0.5	Draft Review	06/27/2023	Gabi Urrutia
1.0	Remediation Plan	06/30/2023	Alejandro Taibo
1.1	Remediation Plan Updates	07/04/2023	Alejandro Taibo
1.2	Remediation Plan Review	07/04/2023	Gokberk Gulgun
1.3	Remediation Plan Review	07/04/2023	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
John Saigle	Halborn	John.Saigle@halborn.com
Alejandro Taibo	Halborn	Alejandro Taibo@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

StaderLabs engaged Halborn to conduct a security audit on their node repository beginning on June 1st, 2023 and ending on June 27th, 2023. The security assessment was scoped to the repository provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks and three days for the engagement and assigned two full-time security engineers to audit the security of the node implementation. The security engineers are blockchain and smart-contract security experts who are skilled in advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that the node implementation functions as intended.
- Identify potential security issues with the node.

In summary, Halborn identified some minor security risks that were partially solved by the StaderLabs team.

1.3 SCOPE

IN-SCOPE CODE & COMMIT:

- Repository: stader-node-v1.1
  - Commit ID: fccb4d64335a439119a0e9a82b290d0c53c14fb6

Previous repository code can also be found in the following public repository:

- Repository: stader-node
  - Commit ID: fccb4d64335a439119a0e9a82b290d0c53c14fb6

REMEDIATION COMMITS:

- Repository: stader-node
  - Commit IDs:
    - 631c4f2850fdf506a2d829f5d1cd1468fa3d18aa
    - 805f4ece2558e3d86817f61aad3dd666854fcfcf

1.4 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of the custom modules. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of structures and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Static Analysis of security for scoped repository, and imported functions. (e.g., staticcheck, gosec, unconvert, codeql, ineffassign and semgrep)
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.
- Dynamic Analysis on files and modules related to the project.
- Custom fuzz testing using Go's built-in fuzzing tools.

2. RISK METHODOLOGY

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

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

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

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

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

2.1 EXPLOITABILITY

Attack Origin (AO):

Captures whether the attack requires compromising a specific account.

Attack Cost (AC):

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

Attack Complexity (AX):

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

Metrics:

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

Exploitability E is calculated using the following formula:

$$E = \prod m_e$$

2.2 IMPACT

Confidentiality (C):

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

Integrity (I):

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

Availability (A):

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

Deposit (D):

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

Yield (Y):

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

Metrics:

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

Impact I is calculated using the following formula:

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

2.3 SEVERITY COEFFICIENT

Reversibility (R):

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

Scope (S):

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

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

Severity Coefficient C is obtained by the following product:

$$C = rs$$

The Vulnerability Severity Score S is obtained by:

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

The score is rounded up to 1 decimal places.

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

3. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	4	13

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
POTENTIAL INTEGER OVERFLOW FOR CYCLES IN CLAIM REWARDS	Low (3.9)	RISK ACCEPTED
POTENTIAL INTEGER OVERFLOW FOR LARGE NONCES AND BLOCK NUMBERS	Low (3.9)	RISK ACCEPTED
USE OF ARCHITECTURE-SPECIFIC INTEGER TYPE COULD CAUSE PROBLEMS ON 32 BIT SYSTEMS	Low (3.9)	NOT APPLICABLE
PRECISION LOSS DUE TO DIVISION OPERATION OCCURRING BEFORE MULTIPLICATION	Low (3.1)	RISK ACCEPTED
FILES AND FOLDERS CREATED BY THE VALIDATOR DO NOT FOLLOW THE PRINCIPLE OF LEAST PRIVILEGE	Informational (1.5)	ACKNOWLEDGED
USE OF UNSUPPORTED GO VERSION	Informational (0.0)	SOLVED - 07/04/2023
USE OF VULNERABLE DEPENDENCIES	Informational (0.0)	ACKNOWLEDGED
ERROR MESSAGE REPORTS INCORRECT CRYPTOGRAPHIC ALGORITHM	Informational (0.0)	SOLVED - 07/04/2023
FUNCTION DECRYPTUSINGPUBLICKEY USES PRIVATEKEY INSTEAD OF PUBLICKEY AS A PARAMETER	Informational (0.0)	SOLVED - 06/30/2023
FUNCTION DECRYPTUSINGPUBLICKEY IS UNUSED	Informational (0.0)	SOLVED - 06/30/2023
TODOS IN CODEBASE	Informational (0.0)	SOLVED - 06/30/2023
USE OF FORMATTING SYMBOL IN PRINTLN	Informational (0.0)	SOLVED - 06/30/2023
VOLUNTARY EXIT MESSAGES CAN EXPIRE	Informational (0.0)	ACKNOWLEDGED
USE TLS IN LISTENER INSTEAD OF PLAIN HTTP	Informational (0.0)	ACKNOWLEDGED
ITERATION OVER A MAP MAY CAUSE ISSUES WITH VALIDATOR STORAGE	Informational (0.0)	ACKNOWLEDGED

FLOATING POINT ARITHMETIC IS NON-DETERMINISTIC	Informational (0.0)	ACKNOWLEDGED
UNHANDLED ERRORS	Informational (0.0)	ACKNOWLEDGED

FINDINGS & TECH DETAILS

4.1 (HAL-01) POTENTIAL INTEGER OVERFLOW FOR CYCLES IN CLAIM REWARDS - LOW (3.9)

Description:

The value for the variable cycle is parsed from a string into an unsigned integer and then converted to a signed integer without any validation. For very large values, this can cause logical issues, as the conversion will result in a negative int64 value.

Code Location:

stader-cli/node/claim-sp-rewards.go

Listing 1: An unchecked type conversion occurs in the else block following the CLI prompt

```
99          cycleSelection := cliutils.Prompt("Select the cycles for
 ↳ which you wish to claim the rewards. Enter the cycles numbers in a
 ↳  comma separate format without any space (e.g. 1,2,3,4) or leave
 ↳ it blank to claim all cycles at once.", "^$|^\\d+(,\\d+)*$", "
 ↳ Unexpected input. Please enter a comma separated list of cycle
 ↳ numbers or leave it blank to claim all cycles at once.")
100         if cycleSelection == "" {
101             for _, cycle := range cycleIndexes {
102                 cyclesToClaim[cycle.Int64()] = true
103             }
104             break
105         } else {
106             elements := strings.Split(cycleSelection, ",")
107             allValid := true
108             for _, element := range elements {
109                 cycle, err := strconv.ParseUint(element, 0, 64)
110                 if err != nil {
111                     fmt.Printf("Unable to parse element: %s",
 ↳ element)
112                     allValid = false
113                 }
```

```
114
115                 // check if unclaimedCycles contains the cycle
116                 found := false
117                 for _, unclaimedCycle := range cycleIndexes {
118                     if unclaimedCycle.Int64() == int64(cycle) {
119                         found = true
120                         break
121                     }
122                 }
123                 if !found {
124                     fmt.Printf("Cycle %d is not in the list of
 ↳ unclaimed cycles. Please enter a valid cycle number\n", cycle)
125                     allValid = false
126                 } else {
127                     cyclesToClaim[int64(cycle)] = true
128                 }
129             }
130
131             if allValid {
132                 break
133             }
134         }
135
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:L/A:L/D:N/Y:N/R:N/S:C (3.9)

Recommendation:

Consider using a single integer type to represent values in order to avoid issues that can occur when converting between types. Alternatively, add validation checks to ensure that the values do not lie outside the range of the `int64` data type. Consider rejecting negative integer values if they do not have a specific use case.

Remediation Plan:

RISK ACCEPTED: The StaderLabs team accepted the risk of this issue and states the following:

Converting cycle to a signed integer is not a concern, as we have confirmed that the value will never exceed the maximum uint64 range. There's no risk of overflow or logical issues in this specific context.

4.2 (HAL-02) POTENTIAL INTEGER OVERFLOW FOR LARGE NONCES AND BLOCK NUMBERS - LOW (3.9)

Description:

Several functions exposed by Ethereum client libraries return uint64 values that are converted to int64 without checking that they will fit into the range of values represented by int64. For very large values, this can cause logical issues, as the conversion will result in a negative int64 value.

Code Location:

stader/api/wallet/status.go

Listing 2: NonceAt() and PendingNonceAt() return uint64 values that are converted to int64

```
31 func getStatus(c *cli.Context) (*api.WalletStatusResponse, error)
↳ {
32
33     // Get services
34     pm, err := services.GetPasswordManager(c)
35     if err != nil {
36         return nil, err
37     }
38     w, err := services.GetWallet(c)
39     if err != nil {
40         return nil, err
41     }
42     ec, err := services.GetEthClient(c)
43     if err != nil {
44         return nil, err
45     }
46
47     // Response
48     response := api.WalletStatusResponse{}
49
```

```
50      // Get wallet status
51      response.PasswordSet = pm.IsPasswordSet()
52      response.WalletInitialized = w.IsInitialized()
53
54      // Get accounts if initialized
55      if response.WalletInitialized {
56
57          // Get node account
58          nodeAccount, err := w.GetNodeAccount()
59          if err != nil {
60              return nil, err
61          }
62          response.AccountAddress = nodeAccount.Address
63
64          currentBlockNumber, err := ec.BlockNumber(context.
↳ Background())
65          if err != nil {
66              return nil, err
67          }
68
69          currentNonce, err := ec.NonceAt(context.Background(),
↳ nodeAccount.Address, big.NewInt(int64(currentBlockNumber)))
70          if err != nil {
71              return nil, err
72          }
73          pendingNonce, err := ec.PendingNonceAt(context.Background
↳ (), nodeAccount.Address)
74          if err != nil {
75              return nil, err
76          }
77
78          response.PendingNonce = big.NewInt(int64(pendingNonce))
79          response.CurrentNonce = big.NewInt(int64(currentNonce))
80      }
81
82      // Return response
83      return &response, nil
84
85  }
```

stader/api/node/claim-sp-rewards.go

Listing 3: GetCurrentBlockNumber() returns a uint64 value that is converted to int64

```
13 func GetCyclesDetailedInfo(c *cli.Context, stringifiedCycles
↳ string) (*api.CyclesDetailedInfo, error) {
14     cfg, err := services.GetConfig(c)
15     if err != nil {
16         return nil, err
17     }
18     sp, err := services.GetSocializingPoolContract(c)
19     if err != nil {
20         return nil, err
21     }
22
23     cycles, err := string_utils.DestringifyArray(stringifiedCycles
↳ )
24     if err != nil {
25         return nil, err
26     }
27
28     response := api.CyclesDetailedInfo{}
29     merkleProofs := []api.DetailedMerkleProofInfo{}
30     for _, cycle := range cycles {
31         merkleCycleProof, exists, err := cfg.StaderNode.
↳ ReadCycleCache(cycle.Int64())
32         if err != nil {
33             return nil, err
34         }
35         if !exists {
36             continue
37         }
38
39         currentBlock, err := eth1.GetCurrentBlockNumber(c)
40         if err != nil {
41             return nil, err
42         }
43         cycleDetails, err := socializing_pool.
↳ GetRewardCycleDetails(sp, cycle, nil)
44         if err != nil {
45             return nil, err
46         }
47
48         cycleStartBlock := currentBlock
49         if cycleDetails.StartBlock.Cmp(big.NewInt(int64(
↳ currentBlock))) < 0 {
```

```
50              cycleStartBlock = cycleDetails.StartBlock.Uint64()
51          }
52          cycleStartTime, err := eth1.ConvertBlockToTimestamp(c,
↳ int64(cycleStartBlock))
53          if err != nil {
54              return nil, err
55          }
56
57          merkleProofs = append(merkleProofs, api.
↳ DetailedMerkleProofInfo{
58              MerkleProofInfo: merkleCycleProof,
59              CycleTime:       cycleStartTime,
60          })
61      }
62
63      response.DetailedCyclesInfo = merkleProofs
64
65      return &response, nil
66  }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:L/A:L/D:N/Y:N/R:N/S:C (3.9)

Recommendation:

Consider using a single integer type to represent values in order to avoid issues that can occur when converting between types. Alternatively, add validation checks to ensure that the values do not lie outside the range of the int64 data type. Consider rejecting negative integer values if they do not have a specific use case.

Remediation Plan:

RISK ACCEPTED: The StaderLabs team accepted the risk of this issue and states the following:

We do not expect nonce or block number to overflow the max of uint64.

4.3 (HAL-03) USE OF ARCHITECTURE-SPECIFIC INTEGER TYPE COULD CAUSE PROBLEMS ON 32 BIT SYSTEMS - LOW (3.9)

Description:

The int type defaults to the system's architecture. On 32-bit systems this may cause unexpected issues as the type conversion to int is equivalent to using int32.

If the program is expecting to always work with 64 bit values, this could lead to integer truncation or overflows on 32-bit systems, which may in turn lead to logic issues.

Code Location:

In this example, the user is prompted for a gwei value. If the number they specify is larger than MAX_INT32, the value will be truncated. This may cause unexpected issues in operation when the value is converted into a float64.

shared/services/gas/gas.go, multiple locations

Listing 4

```
197     fmt.Printf("%s+============== Suggested Gas Prices
 ↳ ==============+\n", log.ColorBlue)
198     fmt.Println("| Avg Wait Time |  Max Fee  |    Total Gas Cost
 ↳    |")
199     fmt.Printf("| %-13s | %-9s | %.4f to %.4f ETH |\n",
200         gasSuggestion.RapidTime, fmt.Sprintf("%d gwei", int(
 ↳ rapidGwei)), rapidLowLimit, rapidHighLimit)
201     fmt.Printf("| %-13s | %-9s | %.4f to %.4f ETH |\n",
202         gasSuggestion.FastTime, fmt.Sprintf("%d gwei", int(
 ↳ fastGwei)), fastLowLimit, fastHighLimit)
203     fmt.Printf("| %-13s | %-9s | %.4f to %.4f ETH |\n",
```

```
204         gasSuggestion.StandardTime, fmt.Sprintf("%d gwei", int(
 ↳ standardGwei)), standardLowLimit, standardHighLimit)
205     fmt.Printf("| %-13s | %-9s | %.4f to %.4f ETH |\n",
206         gasSuggestion.SlowTime, fmt.Sprintf("%d gwei", int(
 ↳ slowGwei)), slowLowLimit, slowHighLimit)
207     fmt.Printf("
 ↳ +================================================+\n\n%s", log.
 ↳ ColorReset)
208
209     fmt.Printf("These prices include a maximum priority fee of %.2
 ↳ f gwei.\n", priorityFee)
210
211     for {
212         desiredPrice := cliutils.Prompt(
213             fmt.Sprintf("Please enter your max fee (including the
 ↳ priority fee) or leave blank for the default of %d gwei:", int(
 ↳ fastGwei)),
214             "^(?:[1-9]\\d*|0)?(?:\\.\\d+)?$",
215             "Not a valid gas price, try again:")
216
217         if desiredPrice == "" {
218             return fastGwei
219         }
220
221         desiredPriceFloat, err := strconv.ParseFloat(desiredPrice,
 ↳  64)
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:L/A:L/D:N/Y:N/R:N/S:C (3.9)

Recommendation:

Consider using int64 explicitly to ensure that no issues occur when the code runs on 32-bit systems.

Remediation Plan:

NOT APPLICABLE: The StaderLabs team states that they do not support 32-bit build. Therefore, this issue is not applicable.

4.4 (HAL-04) PRECISION LOSS DUE TO DIVISION OPERATION OCCURRING BEFORE MULTIPLICATION - LOW (3.1)

Description:

Unless done carefully, division and multiplication operations are typically not commutative when working with floating-point values. Dividing before multiplying can yield a smaller result than multiplying before dividing.

Code Location:

shared/services/gas/gas.go, multiple locations

Listing 5

```
71      if maxFeeGwei != 0 {
72          fmt.Printf("%sUsing the requested max fee of %.2f gwei (
↳ including a max priority fee of %.2f gwei).\n", log.ColorYellow,
↳ maxFeeGwei, maxPriorityFeeGwei)
73
74          var lowLimit float64
75          var highLimit float64
76          if gasLimit == 0 {
77              lowLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasInfo.EstGasLimit)
78              highLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasInfo.SafeGasLimit)
79          } else {
80              lowLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasLimit)
81              highLimit = lowLimit
82          }
83          fmt.Printf("Total cost: %.4f to %.4f ETH%s\n", lowLimit,
↳ highLimit, log.ColorReset)
84
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:L/A:N/D:N/Y:N/R:N/S:C (3.1)

Recommendation:

Perform multiplication before dividing in order to prefer larger values that will not cause calculated amounts to be underestimated.

Remediation Plan:

RISK ACCEPTED: The StaderLabs team accepted the risk of this issue and states the following:

We are not using floating-point vars in any critical steps, for example in gas.go is used for only estimating the gas, to give directional sense for the operators, so it does not need precision.

4.5 (HAL-05) FILES AND FOLDERS CREATED BY THE VALIDATOR DO NOT FOLLOW THE PRINCIPLE OF LEAST PRIVILEGE - INFORMATIONAL (1.5)

Description:

In addition, to the specific vulnerability described above, several folders are configured in the code with lax permissions, such as 775 or 664. In both cases, any user on the operating system can read the contents of folders.

Code Location:

Listing 6

```
stader-cli/service/service.go
543:    err = os.MkdirAll(filepath.Join(volumePath, "validators"),
 0775)

shared/services/stader/client.go
1355:   err = os.Mkdir(runtimeFolder, 0775)
1521:   err = os.MkdirAll(customKeyDir, 0775)

stader/node/node.go
400:            err = os.MkdirAll(validatorsFolder, 0755)
shared/services/stader/client.go
294:    err = os.Chmod(prometheusConfigPath, 0664)
```

BVSS:

AO:S/AC:L/AX:L/C:H/I:N/A:N/D:N/Y:N/R:N/S:U (1.5)

Recommendation:

All folders used by the validator should follow the principle of least privilege. System users should not be able to read the validator's files. Group permissions should be restricted unless there is a key business case for allowing more than one system user to access files used by the node.

Files that do not contain code should not be executable.

Safer file permission strings would be 750 to allow for group access, or 640 for files without code. To restrict files to only the owner, the strings 700 or 600 should be used.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

All the critical files (like validator keys, wallet, etc.)created for stader node do not have read or write access to non-root users.

4.6 (HAL-06) USE OF UNSUPPORTED GO VERSION - INFORMATIONAL (0.0)

Description:

The project uses Go version 1.13. This version has been deprecated. See the Go release notes for their policy on supporting major versions of Go.

Code Location:

go.mod

Listing 7

```
1 module github.com/stader-labs/stader-node
2
3 go 1.13
```

BVSS:

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

Recommendation:

Update to a supported version of Go in order to receive ongoing security updates.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by updating the Go version to 1.16 in the following commit ID:

- 631c4f2850fdf506a2d829f5d1cd1468fa3d18aa.

4.7 (HAL-07) USE OF VULNERABLE DEPENDENCIES - INFORMATIONAL (0.0)

Description:

Several external packages are outdated and/or contain known vulnerabilities.

Code Location:

Excerpts of the output from both tools can be found in the Automated Testing section at the end of the report.

BVSS:

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

Recommendation:

Where possible, keep dependencies patched in order to reduce the risk of the system being attacked using known vulnerabilities. A tool like govulncheck can be added to the project's CI pipeline. This can then be configured to show serious issues that could affect the project.

It is important to note that many of these vulnerabilities flagged by govulncheck are unlikely to be exploitable in practice, as they larger refer to a Web2 context. In addition, the nancy tool reported issues that Halborn determined to be false positives.

Halborn recommends running the nancy and govulncheck tools regularly and to fix as many warnings as possible.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

The code is tested end to end, and the old versions do work as expected , we do not see any risk to the user funds with this.

4.8 (HAL-08) ERROR MESSAGE REPORTS INCORRECT CRYPTOGRAPHIC ALGORITHM - INFORMATIONAL (0.0)

Description:

Error messages relating to cryptographic operations incorrectly state that PKIX public keys are being used.

Code Location:

shared/utils/crypto/rsa.go

Listing 8

```
19 func BytesToPublicKey(pub []byte) (*rsa.PublicKey, error) {
20     block, _ := pem.Decode(pub)
21     b := block.Bytes
22     var err error
23
24     key, err := x509.ParsePKCS1PublicKey(b)
25     if err != nil {
26         fmt.Printf("Error using x509.ParsePKIXPublicKey %v\n", err
↳ )
27         return nil, err
28     }
29
30     return key, nil
31 }
32
33 func BytesToPrivateKey(pub []byte) (*rsa.PrivateKey, error) {
34     block, _ := pem.Decode(pub)
35     b := block.Bytes
36     var err error
37
38     key, err := x509.ParsePKCS1PrivateKey(b)
39     if err != nil {
40         fmt.Printf("Error using x509.ParsePKIXPublicKey %v\n", err
↳ )
41         return nil, err
```

```
42      }
43
44      return key, nil
45  }
```

BVSS:

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

Recommendation:

Correct the debug message so that it matches the code.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by using PKIX instead in the following commit ID:

- 631c4f2850fdf506a2d829f5d1cd1468fa3d18aa.

4.9 (HAL-09) FUNCTION DECRYPTUSINGPUBLICKEY USES PRIVATEKEY INSTEAD OF PUBLICKEY AS A PARAMETER - INFORMATIONAL (0.0)

Description:

The function parameter used for DecryptUsingPublicKey is, in fact, a private key. This could result in cryptographic errors.

Note that this function is not used in the codebase, and so this finding is considered only an informational issue. However, if this function is used in the codebase in a future release, then this confusion between public and private keys could result in cryptographic issues with serious consequences for the protocol.

Code Location:

shared/utils/crypto/rsa.go, L49

Listing 9

```
49 func DecryptUsingPublicKey(data []byte, privateKey *rsa.PrivateKey
↳ ) ([]byte, error) {
50     exitMsgEncrypted, err := rsa.DecryptOAEP(sha256.New(), rand.
↳ Reader, privateKey, data, nil)
51     if err != nil {
52         return nil, err
53     }
54
55     return exitMsgEncrypted, nil
56 }
```

BVSS:

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

Recommendation:

Delete the function if it is not needed. Otherwise, ensure that the function name, parameters, and are internally consistent. Always carefully verify that cryptographic protocols conform with known best practices.

To improve code readability, it is also recommended to change the variable name from `exitMsgEncrypted` to `exitMsgDecrypted` as this function performs decryption.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by removing the aforementioned function in the following commit ID:

- 805f4ece2558e3d86817f61aad3dd666854fcfcf.

4.10 (HAL-10) FUNCTION DECRYPTUSINGPUBLICKEY IS UNUSED - INFORMATIONAL (0.0)

Description:

The function DecryptUsingPublicKey in the file shared/utils/crypto/rsa.go is unused.

Code Location:

shared/utils/crypto/rsa.go, L49

Listing 10

```
49 func DecryptUsingPublicKey(data []byte, privateKey *rsa.PrivateKey
↳ ) ([]byte, error) {
50     exitMsgEncrypted, err := rsa.DecryptOAEP(sha256.New(), rand.
↳ Reader, privateKey, data, nil)
51     if err != nil {
52         return nil, err
53     }
54
55     return exitMsgEncrypted, nil
56 }
```

BVSS:

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

Recommendation:

Remove unused code from the codebase. This can help improve maintainability.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by removing the aforementioned function in the following commit ID:

- 805f4ece2558e3d86817f61aad3dd666854fcfcf.

4.11 (HAL-11) TODOS IN CODEBASE - INFORMATIONAL (0.0)

Description:

Numerous code comments in the codebase contain **TODO** messages.

Code Location:

Listing 11

```
shared/services/wallet/node.go:173:     // TODO: remove this if Prater ever goes away!
stader-cli/node/withdraw-sd.go:39:      // TODO - bchain - rework error messages
stader-cli/service/configMonitoring.go:36:  // TODO:
stader-cli/service/configMonitoring.go:77:  // TODO:
stader-cli/service/configConsensus.go:179:  // Nimbus TODO? check
stader-cli/service/service.go:781:// TODO: this is temporary and can change, clean it up when Nimbus supports split mode
shared/services/config/stader-config.go:800:    // TODO - we have to pick this from stader config but ethx address shouldnt change
shared/utils/validator/fee-recipient.go:169:            // TODO: return here if the container doesn't exist? Is erroring out necessary?
```

BVSS:

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

Recommendation:

It is recommended to use a separate issue tracker or other task management software to track bugs and features rather than using code comments. Developer notes in comments are very likely to be overlooked and to become out of date relative to the code.

If the source code is shared publicly, such developer notes indicate areas of confusion or complexity which may be leveraged by an attacker reading the code.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by removing the aforementioned TODOs comments in the following commit ID:

- 805f4ece2558e3d86817f61aad3dd666854fcfcf.

4.12 (HAL-12) USE OF FORMATTING SYMBOL IN PRINTLN - INFORMATIONAL (0.0)

Description:

A formatting symbol is used in fmt.Println. fmt.Printf or similar should be used to properly display data.

Code Location:

shared/services/gas/gas.go, L223

Listing 12

```
223             fmt.Println("Not a valid gas price (%s), try again.",
 ↳ err.Error())
```

BVSS:

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

Recommendation:

Use fmt.Printf instead of fmt.Println.

Remediation Plan:

SOLVED: The StaderLabs team solved the issue by using fmt.Printf instead of fmt.Println in the following commit ID:

- 805f4ece2558e3d86817f61aad3dd666854fcfcf.

4.13 (HAL-13) VOLUNTARY EXIT MESSAGES CAN EXPIRE - INFORMATIONAL (0.0)

Description:

Periodically, the node executes a routine to verify whether all validators associated to it have their voluntary exit messages registered in the protocol back-end. These voluntary exit messages, which are signed for each validator, could be used later in order to control when a node should be working or not, and in case a node is misbehaving, remove it from the process of validating block.

However, since these signed messages include the fork version, they expire after **two** network upgrades. Therefore, these registered messages should be updated in the back-end under the aforementioned circumstances.

Code Location:

stader/node/node.go, L230

Listing 13: stader/node/node.go (Line 230)

```
225     registeredPresign, ok := preSignRegisteredMap[validatorPubKey.
 ↳ String()]
226     if !ok {
227         errorLog.Printf("Could not query presign api to check if
 ↳ validator: %s is registered\n", validatorPubKey)
228         continue
229     }
230     if registeredPresign {
231         infoLog.Printf("Validator pub key: %s pre signed key
 ↳ already registered\n", validatorPubKey)
232         continue
233     } else {
234         infoLog.Printf("Validator pub key: %s pre signed key not
 ↳ registered. Creating presigned message\n", validatorPubKey)
235     }
```

BVSS:

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

Recommendation:

It is convenient to handle this situation by implementing a way to renew these signed messages and store them in the back-end.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

We are planning to resend pre-sign messages after a hard fork.

4.14 (HAL-14) USE TLS IN LISTENER INSTEAD OF PLAIN HTTP - INFORMATIONAL (0.0)

Description:

It has been identified that the node exports some data containing metrics via HTTP. It is recommended to use TLS for this purpose.

Code Location:

stader/guardian/metrics-exporter.go, L100

Listing 14: (Line 100)

```
 90 http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request)
 ↳ {
 91     w.Write([]byte(`<html>
 92             <head><title>Stader Guardian Metrics Exporter</title></
 ↳ head>
 93             <body>
 94             <h1>Stader Guardian Metrics Exporter</h1>
 95             <p><a href='` + metricsPath + `'>Metrics</a></p>
 96             </body>
 97             </html>`,
 98     ))
 99 })
100 err = http.ListenAndServe(fmt.Sprintf("%s:%d", metricsAddress,
 ↳ metricsPort), nil)
```

BVSS:

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

Recommendation:

It is recommended to use TLS by modifying the highlighted line to http.ListenAndServeTLS which allows TLS usage.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

The metrics server is only used by the local prometheus server as of now. Also, the metrics server runs as a docker container, to serve it externally with SSL the operator can set up SSL certificates with the instance host name

4.15 (HAL-15) ITERATION OVER A MAP MAY CAUSE ISSUES WITH VALIDATOR STORAGE - INFORMATIONAL (0.0)

Description:

Iterating over maps is non-deterministic in Go. As a result, errors can occur when the iteration executes for only some elements of the map. This can happen when the iteration terminates early, e.g., when an error occurs and the code returns before iterating over the remaining elements. This can be especially problematic in a blockchain context when consensus depends on a non-deterministic operation. If the nodes fail to reach the same state due to non-determinism, the chain can halt.

During the engagement, Halborn identified numerous locations where this pattern occurs and may have undesirable side effects. Each of these patterns has been individually checked, and they do not pose a risk in this context.

Code Location:

Due to non-deterministic iteration, the following sections of code could result in undesired system states.

In the first example, the iteration may delete some key stores but not others. For example, if there are 3 key stores, `{A, B, C}`, and only key store `C` triggers an error when `os.RemoveAll` is called, this could result in the following scenarios, depending on the iteration order:

- `C` is evaluated first: No key stores are deleted.
- `C` is evaluated second: Either `A` or `B` is deleted, but not both; `C` is not deleted.
- `C` is evaluated last: `A` and `B` are deleted.

Similar scenarios can occur for all excerpts below.

shared/services/wallet/validator.go

Listing 15

```
181 // Deletes all of the keystore directories and persistent VC
 ↳ storage
182 func (w *Wallet) DeleteValidatorStores() error {
183
184     for name := range w.keystores {
185         keystorePath := w.keystores[name].GetKeystoreDir()
186         err := os.RemoveAll(keystorePath)
187         if err != nil {
188             return fmt.Errorf("error deleting validator directory
 ↳ for %s: %w", name, err)
189         }
190     }
191
192     return nil
193 }
```

Listing 16

```
245 // Save a validator key
246 func (w *Wallet) SaveValidatorKey(key ValidatorKey) error {
247
248     // Update account index
249     if key.WalletIndex > w.ws.NextAccount {
250         w.ws.NextAccount = key.WalletIndex
251     }
252
253     // Update keystores
254     for name := range w.keystores {
255         // Update the keystore in the wallet - using an iterator
 ↳ variable only runs it on the local copy
256         if err := w.keystores[name].StoreValidatorKey(key.
 ↳ PrivateKey, key.DerivationPath); err != nil {
257             return fmt.Errorf("could not store validator key %s in
 ↳  %s keystore: %w", key.PublicKey.Hex(), name, err)
258         }
259     }
260
261     // Return
262     return nil
263
```

```
264 }
```

Listing 17

```
266 // Recover a validator key by public key
267 func (w *Wallet) RecoverValidatorKey(pubkey stadertypes.
 ↳ ValidatorPubkey, startIndex uint) (uint, error) {
268
269     // Check wallet is initialized
270     if !w.IsInitialized() {
271         return 0, errors.New("Wallet is not initialized")
272     }
273
274     // Find matching validator key
275     var index uint
276     var validatorKey *eth2types.BLSPrivateKey
277     var derivationPath string
278     for index = 0; index < MaxValidatorKeyRecoverAttempts; index++
 ↳  {
279         if key, path, err := w.getValidatorPrivateKey(index +
 ↳ startIndex); err != nil {
280             return 0, err
281         } else if bytes.Equal(pubkey.Bytes(), key.PublicKey().
 ↳ Marshal()) {
282             validatorKey = key
283             derivationPath = path
284             break
285         }
286     }
287
288     // Check validator key
289     if validatorKey == nil {
290         return 0, fmt.Errorf("Validator %s key not found", pubkey.
 ↳ Hex())
291     }
292
293     // Update account index
294     nextIndex := index + startIndex + 1
295     if nextIndex > w.ws.NextAccount {
296         w.ws.NextAccount = nextIndex
297     }
298
299     // Update keystores
300     for name := range w.keystores {
```

```
301         // Update the keystore in the wallet - using an iterator
 ↳ variable only runs it on the local copy
302         if err := w.keystores[name].StoreValidatorKey(validatorKey
 ↳ , derivationPath); err != nil {
303             return 0, fmt.Errorf("Could not store %s validator key
 ↳ : %w", name, err)
304         }
305     }
306
307     // Return
308     return index + startIndex, nil
309
310 }
```

stader-cli/service/service.go

Listing 18

```
422 // Updates a configuration from the provided CLI arguments
 ↳ headlessly
423 func configureHeadless(c *cli.Context, cfg *config.StaderConfig)
 ↳ error {
424
425     // Root params
426     for _, param := range cfg.GetParameters() {
427         err := updateConfigParamFromCliArg(c, "", param, cfg)
428         if err != nil {
429             return err
430         }
431     }
432
433     // Subconfigs
434     for sectionName, subconfig := range cfg.GetSubconfigs() {
435         for _, param := range subconfig.GetParameters() {
436             err := updateConfigParamFromCliArg(c, sectionName,
 ↳ param, cfg)
437             if err != nil {
438                 return err
439             }
440         }
441     }
442
443     return nil
```

```
444
445 }
```

BVSS:

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

Recommendation:

When iterating over a map, explicitly sort the results in order to ensure a deterministic operation. Avoid returning early from a loop or aborting on errors in any other way. This can help to prevent cases where execution occurs for only some map elements rather than all of them.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

Stader node doesn't run in a blockchain context so having non deterministic code is not an issue.

4.16 (HAL-16) FLOATING POINT ARITHMETIC IS NON-DETERMINISTIC - INFORMATIONAL (0.0)

Description:

Floating-point arithmetic is often non-deterministic on different machines. Using non-deterministic operations in a blockchain context may result in a chain halt if different validators cannot reach a shared state.

However, this issue does not pose a risk in this component since it doesn't run in a blockchain context.

Code Location:

shared/services/gas/gas.go, multiple locations

Listing 19

```
71      if maxFeeGwei != 0 {
72          fmt.Printf("%sUsing the requested max fee of %.2f gwei (
↳ including a max priority fee of %.2f gwei).\n", log.ColorYellow,
↳ maxFeeGwei, maxPriorityFeeGwei)
73
74          var lowLimit float64
75          var highLimit float64
76          if gasLimit == 0 {
77              lowLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasInfo.EstGasLimit)
78              highLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasInfo.SafeGasLimit)
79          } else {
80              lowLimit = maxFeeGwei / eth.WeiPerGwei * float64(
↳ gasLimit)
81              highLimit = lowLimit
82          }
83          fmt.Printf("Total cost: %.4f to %.4f ETH%s\n", lowLimit,
↳ highLimit, log.ColorReset)
```

```
84
```

BVSS:

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

Recommendation:

Consider refactoring the code to perform operations using integer types rather than floating-point types. For assets such as Ether, arithmetic operations should be calculated using integers that represent wei, the smallest denomination of Ether.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

Stader node doesn't run in the context of a blockchain so having non deterministic code is not an issue.

4.17 (HAL-17) UNHANDLED ERRORS - INFORMATIONAL (0.0)

Description:

Functions called in the code base may return errors that are unchecked. This could lead to undesirable system states where execution occurs on invalid data.

However, each case has been independently checked and none of them poses a risk.

Code Location:

Listing 20: Output from errcheck scan

```
shared/services/beacon/client/std-http-client.go:836:3: _ =
↳ response.Body.Close()
shared/services/beacon/client/std-http-client.go:866:3: _ =
↳ response.Body.Close()
shared/services/config/stader-config.go:459:22: cfg.
↳ applyAllDefaults()
shared/services/gas/etherchain/etherchain.go:68:3:  _ = response.
↳ Body.Close()
shared/services/gas/etherscan/etherscan.go:58:3:    _ = response.
↳ Body.Close()
shared/services/stader/client.go:88:28: ip4Consensus.UseIPProtocol
↳ (4)
shared/services/stader/client.go:95:28: ip6Consensus.UseIPProtocol
↳ (6)
shared/services/stader/client.go:165:3: _ = c.client.Close()
shared/services/stader/client.go:275:12:    os.Setenv(varName,
↳ varValue)
shared/services/stader/client.go:286:12:    os.Setenv(name, value)
shared/services/stader/client.go:392:21:    convertUintParam(param
↳ , &cfg.Lighthouse.MaxPeers, network, 16)
shared/services/stader/client.go:394:21:    convertUintParam(param
↳ , &cfg.Nimbus.MaxPeers, network, 16)
shared/services/stader/client.go:396:21:    convertUintParam(param
↳ , &cfg.Prysm.MaxPeers, network, 16)
```

```
14 shared/services/stader/client.go:398:21:    convertUintParam(param
↳ , &cfg.Teku.MaxPeers, network, 16)
15 shared/services/stader/client.go:401:20:    convertUintParam(param
↳ , &cfg.ConsensusCommon.P2pPort, network, 16)
16 shared/services/stader/client.go:415:20:    convertUintParam(param
↳ , &cfg.Prysm.RpcPort, network, 16)
17 shared/services/stader/client.go:428:20:    convertUintParam(param
↳ , &cfg.BnMetricsPort, network, 16)
18 shared/services/stader/client.go:430:20:    convertUintParam(param
↳ , &cfg.VcMetricsPort, network, 16)
19 shared/services/stader/client.go:432:20:    convertUintParam(param
↳ , &cfg.NodeMetricsPort, network, 16)
20 shared/services/stader/client.go:434:20:    convertUintParam(param
↳ , &cfg.ExporterMetricsPort, network, 16)
21 shared/services/stader/client.go:436:20:    convertUintParam(param
↳ , &cfg.Prometheus.Port, network, 16)
22 shared/services/stader/client.go:438:20:    convertUintParam(param
↳ , &cfg.Grafana.Port, network, 16)
23 shared/services/stader/client.go:479:22:    c.migrateCcSelection(
↳ legacyCfg.Chains.Eth2.Client.Selected, &cfg.Native.ConsensusClient
↳ )
24 shared/services/stader/client.go:517:3: _ = cmd.Close()
25 shared/services/stader/client.go:548:8: _, _ = c.Println(scanner.
↳ Text())
26 shared/services/stader/client.go:1197:21:   convertUintParam(param
↳ , &geth.CacheSize, network, 0)
27 shared/services/stader/client.go:1201:21:   convertUintParam(param
↳ , &geth.MaxPeers, network, 16)
28 shared/services/stader/client.go:1205:21:   convertUintParam(param
↳ , &ecCommon.P2pPort, network, 16)
29 shared/services/stader/client.go:1364:12:   os.Setenv(varName,
↳ varValue)
30 shared/services/stader/client.go:1369:13:   os.Setenv(name, value)
31 shared/services/stader/client.go:1575:13:   os.Setenv(key,
↳ shellescape.Quote(value))
32 shared/services/stader/client.go:1737:17:   defer cmd.Close()
33 shared/services/stader/client.go:1766:3:    _ = cmd.Close()
34 shared/utils/cli/prompt.go:76:9:    index, _ := strconv.Atoi(
↳ response)
35 shared/utils/stader/merkle-proof-download.go:24:22: defer res.Body
↳ .Close()
36 shared/utils/stader/pre-signed-flows.go:24:22:  defer res.Body.
↳ Close()
37 shared/utils/stader/pre-signed-flows.go:45:22:  defer res.Body.
```

```
↳ Close()
38 shared/utils/stader/pre-signed-flows.go:69:22:  defer res.Body.
↳ Close()
39 shared/utils/stader/pre-signed-flows.go:86:22:  defer res.Body.
↳ Close()
40 shared/utils/stader/pre-signed-flows.go:107:22: defer res.Body.
↳ Close()
41 shared/utils/validator/fee-recipient.go:56:15:  clientType, _ :=
↳ bc.GetClientType()
42 shared/utils/validator/fee-recipient.go:137:15: clientType, _ :=
↳ bc.GetClientType()
43 stader-cli/service/service.go:620:27:   staderClient.SaveConfig(
↳ cfg)
44 stader-cli/validator/export.go:58:18:   defer file.Close()
45 stader-cli/wallet/init.go:108:2:    _ = term.Clear()
46 stader-lib/stader/abi.go:47:3:  _ = zlibReader.Close()
47 stader/guardian/metrics-exporter.go:91:10:  w.Write([]byte(`<html>
```

BVSS:

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

Recommendation:

Always handle errors safely to avoid unexpected negative outcomes.

Remediation Plan:

ACKNOWLEDGED: The StaderLabs team acknowledged this issue and states the following:

There is no risk posed (such as a node crash or loss of funds)by not handling the errors at the specified places.

AUTOMATED TESTING

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped component. Among the tools used were staticcheck, gosec, semgrep, unconvert, CodeQL and Nancy. After Halborn verified all the 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 21: 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
```

AUTOMATED TESTING

AUTOMATED TESTING

```
Scan Status

Scanning 319 files with 1077 Code rules:

Language      Rules   Files          Origin      Rules
<multilang>      60     879          Community    1077
go               86     218
bash              4      19
json              4      14
yaml             27      13
dockerfile        4       1

                                    100% 0:00:02


28 Code Findings

  shared/services/beacon/client/std-http-client.go
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `attestationInfo[i]` be modified when an error could be returned?
        Details: https://sg.run/qq6y

        502┆ attestationInfo[i].AggregationBits, err = hex.DecodeString(bitString)
          ┆----------------------------------------
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `info` be modified when an error could be returned?
        Details: https://sg.run/qq6y

        541┆ info.AggregationBits, err = hex.DecodeString(bitString)

  shared/services/config/stader-config.go
     javascript.lang.security.detect-insecure-websocket.detect-insecure-websocket
        Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket
        connections.
        Details: https://sg.run/GWyz

        817┆ envVars["EC_WS_ENDPOINT"] = fmt.Sprintf("ws://%s:%d", Eth1ContainerName,
cfg.ExecutionCommon.WsPort.Value)
          ┆----------------------------------------
        819┆ envVars["EC_ENGINE_WS_ENDPOINT"] = fmt.Sprintf("ws://%s:%d", Eth1ContainerName,
cfg.ExecutionCommon.EnginePort.Value)
          ┆----------------------------------------
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `cfg` be modified when an error could be returned?
        Details: https://sg.run/qq6y

        771┆ cfg.IsNativeMode, err = strconv.ParseBool(masterMap[rootConfigName]["isNative"])

  shared/services/stader/command.go
     go.lang.security.audit.dangerous-exec-command.dangerous-exec-command
        Detected non-static command inside Command. Audit the input to 'exec.Command'. If unverifi
        user data can reach this call site, this is a code injection vulnerability. A malicious
        actor can inject a malicious script to execute arbitrary code.
        Details: https://sg.run/W8lA

         40┆ cmd:     exec.Command("sh", "-c", cmdText),
```

Semgrep Results

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

         66┆ m.BeaconConfig, err = m.bc.GetEth2Config()

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

        234┆ w.mk, err = hdkeychain.NewMaster(w.seed, &chaincfg.MainNetParams)
           ┆----------------------------------------
        375┆ w.seed, err = w.encryptor.Decrypt(w.ws.Crypto, password)
           ┆----------------------------------------
        381┆ w.mk, err = hdkeychain.NewMaster(w.seed, &chaincfg.MainNetParams)
           ┆----------------------------------------
        399┆ w.mk, err = hdkeychain.NewMaster(w.seed, &chaincfg.MainNetParams)

  shared/types/config/parameter.go
     trailofbits.go.questionable-assignment.questionable-assignment
        Should `param` be modified when an error could be returned?
        Details: https://sg.run/qq6y

        103┆ param.Value, err = strconv.ParseInt(value, 0, 0)
           ┆----------------------------------------
        105┆ param.Value, err = strconv.ParseUint(value, 0, 0)
           ┆----------------------------------------
        111┆ param.Value, err = strconv.ParseBool(value)
```

```
          56┆ response.SocializingPoolContract, err = services.GetSocializingPoolAddress(c)
            ┆----------------------------------------
          60┆ response.PermisionlessPool, err = services.GetPermissionlessPoolAddress(c)
            ┆----------------------------------------
          64┆ response.StaderOracle, err = services.GetStaderOracleAddress(c)
            ┆----------------------------------------
          68┆ response.StakePoolManager, err = services.GetStakePoolManagerAddress(c)

   stader/api/wallet/recover.go
      trailofbits.go.questionable-assignment.questionable-assignment
         Should `response` be modified when an error could be returned?
         Details: https://sg.run/qq6y

          89┆ response.ValidatorKeys, err = walletutils.RecoverStaderKeys(pnr, nodeAccount.Address, w, false)

   stader/guardian/metrics-exporter.go
      go.lang.security.audit.net.use-tls.use-tls
         Found an HTTP server without TLS. Use 'http.ListenAndServeTLS' instead. See
         https://golang.org/pkg/net/http/#ListenAndServeTLS for more information.
         Details: https://sg.run/dKbY

          ▶▶┆ Autofix ▶ http.ListenAndServeTLS(fmt.Sprintf("%s:%d", metricsAddress, metricsPort),
certFile, keyFile, nil)
         100┆ err = http.ListenAndServe(fmt.Sprintf("%s:%d", metricsAddress, metricsPort), nil)


┌──────────────┐
│ Scan Summary │
└──────────────┘
Some files were skipped or only partially analyzed.
  Partially scanned: 1 files only partially analyzed due to parsing or internal Semgrep errors

Ran 1077 rules on 293 files: 28 findings.
```

- No major issues found by Semgrep.

AUTOMATED TESTING

Gosec:

```
[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/guardian/metrics-exporter.go:100] - G114 (CWE-676): Use of net/http serve
ce: HIGH, Severity: MEDIUM)
   99:         })
 > 100:        err = http.ListenAndServe(fmt.Sprintf("%s:%d", metricsAddress, metricsPort), nil)
   101:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/utils/validator/fee-recipient.go:197] - G204 (CWE-78): Subprocess launche
   196:                // Run validator stop command bound to os stdout/stderr
 > 197:                cmd := exec.Command(stopCommand)
   198:                cmd.Stdout = os.Stdout


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/utils/validator/fee-recipient.go:108] - G204 (CWE-78): Subprocess launche
   107:                // Run validator restart command bound to os stdout/stderr
 > 108:                cmd := exec.Command(restartCommand)
   109:                cmd.Stdout = os.Stdout


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/node/merkle-proofs-download.go:76] - G304 (CWE-22): Potential file inclus
   75:                 m.log.Println("Downloading merkle proof for cycle %d", cycleMerkleProof.Cycle)
 > 76:                 file, err := os.Create(absolutePathOfProofFile)
   77:                 if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/api/node/download-sp-merkle-proofs.go:106] - G304 (CWE-22): Potential fil
)
   105:
 > 106:                file, err := os.Create(absolutePathOfProofFile)
   107:                if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader-cli/wallet/utils.go:160] - G304 (CWE-22): Potential file inclusion via va
   159:                // Read the file
 > 160:                bytes, err := ioutil.ReadFile(filepath.Join(customKeyDir, file.Name()))
   161:                if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/wallet/keystore/prysm/keystore.go:217] - G304 (CWE-22): Potentia
EDIUM)
   216:        // Get the random keystore password
 > 217:        passwordBytes, err := ioutil.ReadFile(passwordFilePath)
   218:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/wallet/keystore/prysm/keystore.go:120] - G304 (CWE-22): Potentia
EDIUM)
   119:        passwordFilePath := filepath.Join(ks.keystorePath, KeystoreDir, WalletDir, AccountsDir, KeystorePasswordFileName)
 > 120:        passwordBytes, err := ioutil.ReadFile(passwordFilePath)
   121:        if err != nil {
```

Security Analysis Output Sample

```
[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/legacy-client.go:69] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   68:         }
 > 69:         configBytes, err := ioutil.ReadFile(expandedPath)
   70:         if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/fee-recipient.go:52] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   51:         expectedString := getFeeRecipientFileContents(feeRecipient, cfg)
 > 52:         bytes, err := ioutil.ReadFile(path)
   53:         if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/config/stadernode-config.go:518] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   517:        // Open the JSON file
 > 518:        file, err := os.Open(absolutePathOfProofFile)
   519:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/config/stadernode-config.go:453] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   452:                }
 > 453:                data, err := os.ReadFile(expandedCycleMerkleRewardFile)
   454:                if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/config/stader-config.go:147] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   146:        // Read the file
 > 147:        configBytes, err := ioutil.ReadFile(path)
   148:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/config/config-legacy.go:304] - G304 (CWE-22): Potential file inclusion via variable (Confidence: HIGH, Severity: MEDIUM)
   303:        // Read file; squelch not found errors if file is optional
 > 304:        bytes, err := ioutil.ReadFile(path)
   305:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:294] - G302 (CWE-276): Expect file permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   293:        }
 > 294:        err = os.Chmod(prometheusConfigPath, 0664)
   295:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/node/node.go:400] - G301 (CWE-276): Expect directory permissions to be 0750 or less (Confidence: HIGH, Severity: MEDIUM)
   399:                validatorsFolder := filepath.Dir(feeRecipientPath)
 > 400:                err = os.MkdirAll(validatorsFolder, 0755)
   401:                if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader-cli/service/service.go:543] - G301 (CWE-276): Expect directory permissions to be 0750 or less (Confidence: HIGH, Severity: MEDIUM)
```

AUTOMATED TESTING

```
   542:        fmt.Print("Recreating data folder... ")
 > 543:        err = os.MkdirAll(filepath.Join(volumePath, "validators"), 0775)
   544:        if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1521] - G301 (CWE-276): Expect directory permissions to be 0750 or less (Confidence: HIGH, Severity: MEDIUM)
   1520:       }
 > 1521:       err = os.MkdirAll(customKeyDir, 0775)
   1522:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1355] - G301 (CWE-276): Expect directory permissions to be 0750 or less (Confidence: HIGH, Severity: MEDIUM)
   1354:       }
 > 1355:       err = os.Mkdir(runtimeFolder, 0775)
   1356:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/node/node.go:414] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   413:                }
 > 414:                err := ioutil.WriteFile(feeRecipientPath, []byte(defaultFeeRecipientFileContents), 0664)
   415:                if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/utils/stdr/config.go:61] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   60:
 > 61:         if err := ioutil.WriteFile(path, configBytes, 0664); err != nil {
   62:                 return fmt.Errorf("could not write Stader config to %s: %w", shellescape.Quote(path), err)


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1507] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1506:               mevBoostComposePath := filepath.Join(runtimeFolder, config.MevBoostContainerName+composeFileSuffix)
 > 1507:               err = ioutil.WriteFile(mevBoostComposePath, contents, 0664)
   1508:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1492] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1491:               prometheusComposePath := filepath.Join(runtimeFolder, config.PrometheusContainerName+composeFileSuffix)
 > 1492:               err = ioutil.WriteFile(prometheusComposePath, contents, 0664)
   1493:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1479] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1478:               exporterComposePath := filepath.Join(runtimeFolder, config.ExporterContainerName+composeFileSuffix)
 > 1479:               err = ioutil.WriteFile(exporterComposePath, contents, 0664)
   1480:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1466] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1465:               grafanaComposePath := filepath.Join(runtimeFolder, config.GrafanaContainerName+composeFileSuffix)
 > 1466:               err = ioutil.WriteFile(grafanaComposePath, contents, 0664)
   1467:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1450] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1449:               eth2ComposePath := filepath.Join(runtimeFolder, config.Eth2ContainerName+composeFileSuffix)
 > 1450:               err = ioutil.WriteFile(eth2ComposePath, contents, 0664)
   1451:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1435] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1434:               eth1ComposePath := filepath.Join(runtimeFolder, config.Eth1ContainerName+composeFileSuffix)
 > 1435:               err = ioutil.WriteFile(eth1ComposePath, contents, 0664)
   1436:               if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1421] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1420:       validatorComposePath := filepath.Join(runtimeFolder, config.ValidatorContainerName+composeFileSuffix)
 > 1421:       err = ioutil.WriteFile(validatorComposePath, contents, 0664)
   1422:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1408] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1407:       guardianComposePath := filepath.Join(runtimeFolder, config.GuardianContainerName+composeFileSuffix)
 > 1408:       err = ioutil.WriteFile(guardianComposePath, contents, 0664)
   1409:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1395] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1394:       nodeComposePath := filepath.Join(runtimeFolder, config.NodeContainerName+composeFileSuffix)
 > 1395:       err = ioutil.WriteFile(nodeComposePath, contents, 0664)
   1396:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1382] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   1381:       apiComposePath := filepath.Join(runtimeFolder, config.ApiContainerName+composeFileSuffix)
 > 1382:       err = ioutil.WriteFile(apiComposePath, contents, 0664)
   1383:       if err != nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:290] - G306 (CWE-276): Expect WriteFile permissions to be 0600 or less (Confidence: HIGH, Severity: MEDIUM)
   289:        // Write the actual Prometheus config file
 > 290:        err = ioutil.WriteFile(prometheusConfigPath, contents, 0664)
   291:        if err != nil {
```

AUTOMATED TESTING

```
[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader/guardian/metrics-exporter.go:91-98] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   90:         http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
  > 91:             w.Write([]byte(`<html>
  > 92:             <head><title>Stader Guardian Metrics Exporter</title></head>
  > 93:             <body>
  > 94:             <h1>Stader Guardian Metrics Exporter</h1>
  > 95:             <p><a href='` + metricsPath + `'>Metrics</a></p>
  > 96:             </body>
  > 97:             </html>`,
  > 98:                 ))
   99:         })


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/stader-cli/service/service.go:620] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   619:                         }
  > 620:                         staderClient.SaveConfig(cfg)
   621:                         fmt.Printf("%sUpdated settings successfully.%s\n", colorGreen, colorReset)


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1575] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1574:                 for key, value := range envVars {
  > 1575:                         os.Setenv(key, shellescape.Quote(value))
   1576:                         envArgs += fmt.Sprintf("-e %s ", key)


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1369] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1368:                 for name, value := range oldValues {
  > 1369:                         os.Setenv(name, value)
   1370:                 }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1364] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1363:                 oldValues[varName] = os.Getenv(varName)
  > 1364:                 os.Setenv(varName, varValue)
   1365:         }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1205] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1204:                         if ecCommon != nil {
  > 1205:                                 convertUintParam(param, &ecCommon.P2pPort, network, 16)
   1206:                         }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1201] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1200:                         if geth != nil {
  > 1201:                                 convertUintParam(param, &geth.MaxPeers, network, 16)
   1202:                         }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:1197] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   1196:                         if geth != nil {
  > 1197:                                 convertUintParam(param, &geth.CacheSize, network, 0)
   1198:                         }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:479] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   478:         cfg.Native.CcHttpUrl.Value = legacyCfg.Chains.Eth2.Provider
  > 479:         c.migrateCcSelection(legacyCfg.Chains.Eth2.Client.Selected, &cfg.Native.ConsensusClient)
   480:         cfg.Native.ValidatorRestartCommand.Value = legacyCfg.StaderNode.ValidatorRestartCommand


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:438] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   437:                 case "GRAFANA_PORT":
  > 438:                         convertUintParam(param, &cfg.Grafana.Port, network, 16)
   439:                 }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:436] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   435:                 case "PROMETHEUS_PORT":
  > 436:                         convertUintParam(param, &cfg.Prometheus.Port, network, 16)
   437:                 case "GRAFANA_PORT":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:434] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   433:                 case "EXPORTER_METRICS_PORT":
  > 434:                         convertUintParam(param, &cfg.ExporterMetricsPort, network, 16)
   435:                 case "PROMETHEUS_PORT":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:432] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   431:                 case "NODE_METRICS_PORT":
  > 432:                         convertUintParam(param, &cfg.NodeMetricsPort, network, 16)
   433:                 case "EXPORTER_METRICS_PORT":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:430] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   429:                 case "VALIDATOR_METRICS_PORT":
  > 430:                         convertUintParam(param, &cfg.VcMetricsPort, network, 16)
   431:                 case "NODE_METRICS_PORT":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:428] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   427:                 case "ETH2_METRICS_PORT":
  > 428:                         convertUintParam(param, &cfg.BnMetricsPort, network, 16)
   429:                 case "VALIDATOR_METRICS_PORT":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:415] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
   414:                 case "ETH2_RPC_PORT":
  > 415:                         convertUintParam(param, &cfg.Prysm.RpcPort, network, 16)
   416:                         port := cfg.Prysm.RpcPort.Value.(uint16)
```

AUTOMATED TESTING

```
[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:401] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    400:                case "ETH2_P2P_PORT":
  > 401:                        convertUintParam(param, &cfg.ConsensusCommon.P2pPort, network, 16)
    402:                case "ETH2_CHECKPOINT_SYNC_URL":


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:398] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    397:                        case cfgtypes.ConsensusClient_Teku:
  > 398:                                convertUintParam(param, &cfg.Teku.MaxPeers, network, 16)
    399:                        }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:396] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    395:                        case cfgtypes.ConsensusClient_Prysm:
  > 396:                                convertUintParam(param, &cfg.Prysm.MaxPeers, network, 16)
    397:                        case cfgtypes.ConsensusClient_Teku:


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:394] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    393:                        case cfgtypes.ConsensusClient_Nimbus:
  > 394:                                convertUintParam(param, &cfg.Nimbus.MaxPeers, network, 16)
    395:                        case cfgtypes.ConsensusClient_Prysm:


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:392] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    391:                        case cfgtypes.ConsensusClient_Lighthouse:
  > 392:                                convertUintParam(param, &cfg.Lighthouse.MaxPeers, network, 16)
    393:                        case cfgtypes.ConsensusClient_Nimbus:


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:286] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    285:        for name, value := range oldValues {
  > 286:                os.Setenv(name, value)
    287:        }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:275] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    274:                oldValues[varName] = os.Getenv(varName)
  > 275:                os.Setenv(varName, varValue)
    276:        }


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:95] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    94:         ip6Consensus := externalip.DefaultConsensus(nil, nil)
  > 95:         ip6Consensus.UseIPProtocol(6)
    96:         return ip6Consensus.ExternalIP()


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/stader/client.go:88] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    87:         ip4Consensus := externalip.DefaultConsensus(nil, nil)
  > 88:         ip4Consensus.UseIPProtocol(4)
    89:         if ip, err := ip4Consensus.ExternalIP(); err == nil {


[/home/kaorz/Documents/Work/Halborn/Projects/staderlabs/stader-node-v1.1/shared/services/config/stader-config.go:459] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)
    458:        cfg.StaderNode.Network.Value = cfg.StaderNode.Network.Options[0].Value
  > 459:        cfg.applyAllDefaults()
    460:


Summary:
  Gosec  : 2.16.0
  Files  : 216
  Lines  : 70958
  Nosec  : 0
  Issues : 58
```

- File permission issues were flagged correctly
- Flagged potential file inclusion has been reviewed, and it does not pose any risk
- Code execution related issues have been reviewed; they do not pose any risk
- Some unhandled errors were flagged correctly
- No major issues found by gosec

CodeQL:

Severity : warning [4]

- crypto-com/cosmos-sdk-codeql/floating-point-arithmetic Floating point arithmetic operations are not associative and a possible source of non-determinism: **61**
- crypto-com/cosmos-sdk-codeql/map-iteration Iteration over map may be a possible source of non-determinism: **30**
  - hared/services/config/stader-config.go:494
  - hared/services/config/stader-config.go:577
  - hared/services/config/stader-config.go:724
  - hared/services/config/stader-config.go:778
  - hared/services/config/stader-config.go:1001
  - hared/services/config/stader-config.go:1023
  - hared/services/config/stader-config.go:1025
  - hared/services/config/stader-config.go:1099
  - hared/services/config/stader-config.go:1122
  - hared/services/stader/client.go:273
  - hared/services/stader/client.go:285
  - hared/services/stader/client.go:1316
  - hared/services/stader/client.go:1362
  - hared/services/stader/client.go:1368
  - hared/services/stader/client.go:1574
  - hared/services/stader/client.go:1585
  - hared/services/wallet/validator.go:168
  - hared/services/wallet/validator.go:183
  - hared/services/wallet/validator.go:253
  - hared/services/wallet/validator.go:299
  - hared/utils/sys/cpu-flags.go:54
  - tader-cli/node/claim-sp-rewards.go:135
  - tader-cli/service/commands.go:112
  - tader-cli/service/service.go:434
  - tader/api/node/status.go:250
  - tader/guardian/collector/beacon-chain-collector.go:106
  - tader/guardian/collector/beacon-chain-collector.go:125
  - tader/guardian/collector/beacon-chain-collector.go:144
  - tader/guardian/collector/beacon-chain-collector.go:160
  - tader/node/node.go:299
- crypto-com/cosmos-sdk-codeql/sensitive-import Certain system packages contain functions which may be a possible source of non-determinism: **10**
- crypto-com/cosmos-sdk-codeql/goroutine Spawning a Go routine may be a possible source of non-determinism: **9**

Figure 1: Sample of CodeQL results

Security Analysis Output Sample

Govulncheck:

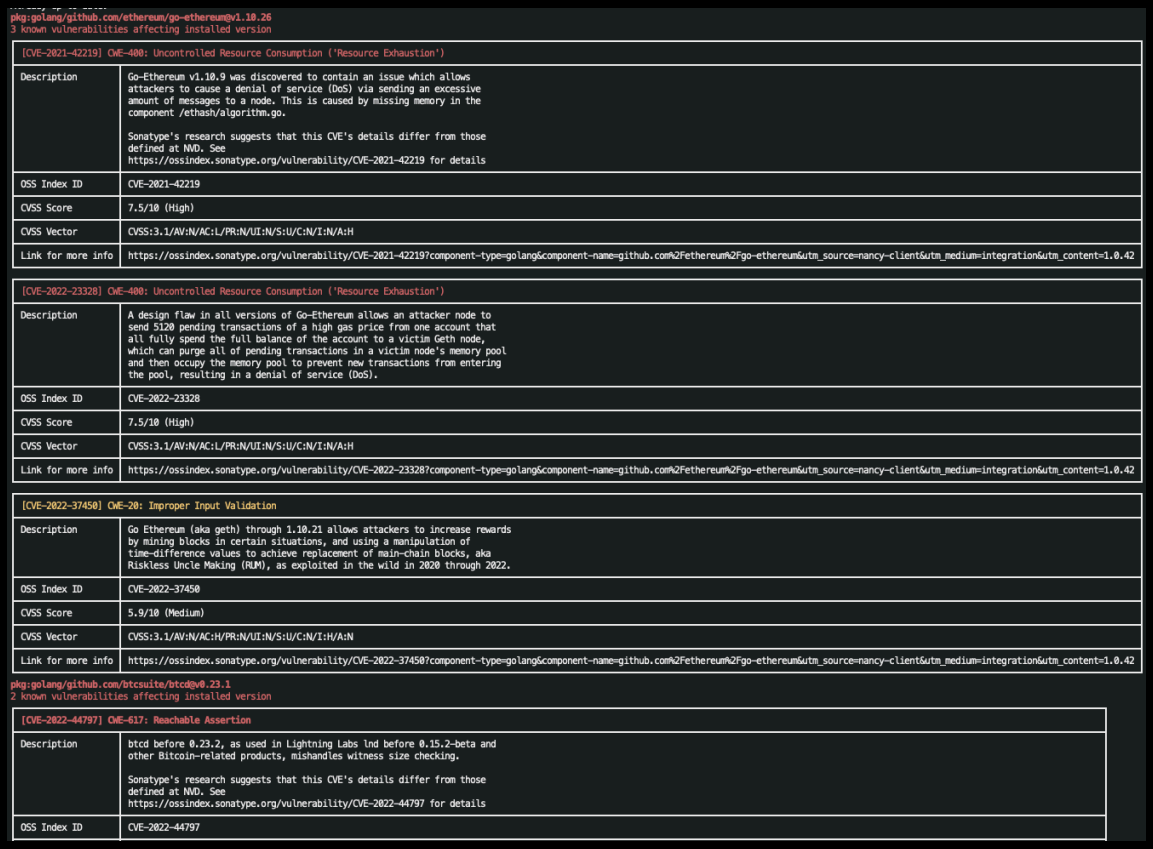
```
Vulnerability #1: GO-2023-1840
  On Unix platforms, the Go runtime does not behave differently
  when a binary is run with the setuid/setgid bits. This can be
  dangerous in certain cases, such as when dumping memory state,
  or assuming the status of standard i/o file descriptors. If a
  setuid/setgid binary is executed with standard I/O file
  descriptors closed, opening any files can result in unexpected
  content being read or written with elevated privileges.
  Similarly, if a setuid/setgid program is terminated, either via
  panic or signal, it may leak the contents of its registers.

  More info: https://pkg.go.dev/vuln/GO-2023-1840

  Standard library
    Found in: runtime@go1.20.3
    Fixed in: runtime@go1.20.5

    Call stacks in your code:
      github.com/stader-labs/stader-node/shared/services.init calls github.com/ethereum/go-ethereum/ethclient.init, which eventually calls runti
      github.com/stader-labs/stader-node/shared/services/beacon/client.init calls github.com/prysmaticlabs/prysm/v3/crypto/bls.init, which event
      github.com/stader-labs/stader-node/shared/utils/sys.init calls runtime.init
      github.com/stader-labs/stader-node/stader-lib/utils/eth.init calls github.com/ethereum/go-ethereum/accounts/abi/bind.init, which eventuall
      github.com/stader-labs/stader-node/stader-lib/utils/eth.init calls github.com/ethereum/go-ethereum/accounts/abi/bind.init, which eventuall
      github.com/stader-labs/stader-node/stader-lib/utils/eth.init calls github.com/ethereum/go-ethereum/accounts/abi/bind.init, which eventuall
      github.com/stader-labs/stader-node/stader-lib/utils/eth.init calls github.com/ethereum/go-ethereum/accounts/abi/bind.init, which eventuall
      github.com/stader-labs/stader-node/stader/guardian.init calls github.com/prometheus/client_golang/prometheus/promhttp.init, which eventual
      github.com/stader-labs/stader-node/stader/guardian/collector.init calls github.com/prometheus/client_golang/prometheus.init, which eventua
      github.com/stader-labs/stader-node/stader/guardian/collector.init calls github.com/prometheus/client_golang/prometheus.init, which eventua
      github.com/stader-labs/stader-node/stader/node.init calls github.com/wealdtech/go-eth2-types/v2.init, which eventually calls runtime.Calle
      shared/services/gas/gas.go:134:132: github.com/stader-labs/stader-node/shared/services/gas.GetHeadlessMaxFeeWei calls runtime.TypeAssertio
      shared/services/gas/gas.go:134:132: github.com/stader-labs/stader-node/shared/services/gas.GetHeadlessMaxFeeWei calls runtime.plainError.E
      shared/services/services.go:536:15: github.com/stader-labs/stader-node/shared/services.getDocker calls sync.Once.Do, which eventually call
      shared/services/services.go:536:15: github.com/stader-labs/stader-node/shared/services.getDocker calls sync.Once.Do, which eventually call
      shared/services/services.go:536:15: github.com/stader-labs/stader-node/shared/services.getDocker calls sync.Once.Do, which eventually call
      shared/utils/log/logger.go:61:13: github.com/stader-labs/stader-node/shared/utils/log.ColorLogger.Println calls log.Println, which eventu
      shared/utils/validator/fee-recipient.go:197:22: github.com/stader-labs/stader-node/shared/utils/validator.StopValidator calls os/exec.Comm
      stader-cli/stader-cli.go:854:19: github.com/stader-labs/stader-node/stader-cli.main calls github.com/urfave/cli.App.Run, which eventually
      stader/guardian/metrics-exporter.go:100:27: github.com/stader-labs/stader-node/stader/guardian.runMetricsServer calls net/http.ListenAndSe
      stader/guardian/metrics-exporter.go:100:27: github.com/stader-labs/stader-node/stader/guardian.runMetricsServer calls net/http.ListenAndSe
      stader/stader.go:825:16: github.com/stader-labs/stader-node/stader.main calls fmt.Fprintln, which e
ventually calls runtime.Func.FileLine
      stader/stader.go:825:16: github.com/stader-labs/stader-node/stader.main calls fmt.Fprintln, which eventually calls runtime.ReadMemStats

=== Informational ===

Found 1 vulnerability in packages that you import, but there are no call
stacks leading to the use of this vulnerability. You may not need to
take any action. See https://pkg.go.dev/golang.org/x/vuln/cmd/govulncheck
for details.

Vulnerability #1: GO-2022-1098
  Erroneous message decoding can cause denial of service. Improper
  checking of maximum witness size during node message decoding
  prevented nodes in Lightning Labs lnd (before 0.15.2-beta) to
  sync.
  More info: https://pkg.go.dev/vuln/GO-2022-1098
  Found in: github.com/btcsuite/btcd@v0.23.1
  Fixed in: github.com/btcsuite/btcd@v0.23.2
```

Security Analysis Output Sample

Nancy:

Security Analysis Output Sample It is important to note that, while Nancy reports issues in the project's dependencies, Halborn verified that all issues reported by Nancy are false positives that do not affect this project:

- go-ethereum vulnerabilties: Nancy reports vulnerabilities that affect older versions of Geth; Stader's version is safe.
- btcsuite vulnerabilities: The codebase uses Bitcoin libraries only for verifying hardware wallets and is unaffected by the vulnerabilities associated with this version.



pkg:golang/github.com/ethereum/go-ethereum@v1.10.26
3 known vulnerabilities affecting installed version

[CVE-2021-42219] CWE-400: Uncontrolled Resource Consumption ('Resource Exhaustion')	
Description	Go-Ethereum v1.10.9 was discovered to contain an issue which allows attackers to cause a denial of service (DoS) via sending an excessive amount of messages to a node. This is caused by missing memory in the component /ethash/algorithm.go. Sonatype's research suggests that this CVE's details differ from those defined at NVD. See https://ossindex.sonatype.org/vulnerability/CVE-2021-42219 for details
OSS Index ID	CVE-2021-42219
CVSS Score	7.5/10 (High)
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
Link for more info	https://ossindex.sonatype.org/vulnerability/CVE-2021-42219?component-type=golang&component-name=github.com%2Fethereum%2Fgo-ethereum&utm_source=nancy-client&utm_medium=integration&utm_content=1.0.42

[CVE-2022-23328] CWE-400: Uncontrolled Resource Consumption ('Resource Exhaustion')	
Description	A design flaw in all versions of Go-Ethereum allows an attacker node to send 5120 pending transactions of a high gas price from one account that all fully spend the full balance of the account to a victim Geth node, which can purge all of pending transactions in a victim node's memory pool and then occupy the memory pool to prevent new transactions from entering the pool, resulting in a denial of service (DoS).
OSS Index ID	CVE-2022-23328
CVSS Score	7.5/10 (High)
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
Link for more info	https://ossindex.sonatype.org/vulnerability/CVE-2022-23328?component-type=golang&component-name=github.com%2Fethereum%2Fgo-ethereum&utm_source=nancy-client&utm_medium=integration&utm_content=1.0.42

[CVE-2022-37450] CWE-20: Improper Input Validation	
Description	Go Ethereum (aka geth) through 1.10.21 allows attackers to increase rewards by mining blocks in certain situations, and using a manipulation of time-difference values to achieve replacement of main-chain blocks, aka Riskless Uncle Making (RUM), as exploited in the wild in 2020 through 2022.
OSS Index ID	CVE-2022-37450
CVSS Score	5.9/10 (Medium)
CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N
Link for more info	https://ossindex.sonatype.org/vulnerability/CVE-2022-37450?component-type=golang&component-name=github.com%2Fethereum%2Fgo-ethereum&utm_source=nancy-client&utm_medium=integration&utm_content=1.0.42

pkg:golang/github.com/btcsuite/btcd@v0.23.1
2 known vulnerabilities affecting installed version

[CVE-2022-44797] CWE-617: Reachable Assertion	
Description	btcd before 0.23.2, as used in Lightning Labs lnd before 0.15.2-beta and other Bitcoin-related products, mishandles witness size checking. Sonatype's research suggests that this CVE's details differ from those defined at NVD. See https://ossindex.sonatype.org/vulnerability/CVE-2022-44797 for details
OSS Index ID	CVE-2022-44797

Figure 2: Sample of Nancy results

THANK YOU FOR CHOOSING

// HALBORN