// HALBORN

DRAFT

Maya - ETH Router

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **November 28th, 2022 - December 13th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	11/28/2022	John Saigle
0.2	Document Updates	12/12/2022	John Saigle
0.3	Draft Review	12/13/2022	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
John Saigle	Halborn	John.Saigle@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Maya engaged Halborn to conduct a security audit on their ETH router smart contract, beginning on November 28th, 2022 and ending on December 13th, 2022. The security assessment was scoped to the modifications of the code related to the Eth Router functionality.

1.2 AUDIT SUMMARY

The team at Halborn was provided several days for the engagement and assigned a full-time security engineer to audit the security of the modifications made to the smart contracts in the ETH Router repository. The security engineer is a blockchain and smart contract security expert with advanced penetration testing, smart contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that the changes to the ETH Router smart contract operate as intended.
- Identify potential security issues with the custom code introduced by Maya.
- Outline any risks that may be created by modifying the existing codebase.
- Verify the use of Solidity best practices.

In summary, Halborn identified some security risks that should be addressed by the Maya team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard

to the scope of the custom 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.
- Manual Assessment for discovering security vulnerabilities on codebase.
- Ensuring correctness of the codebase.
- Dynamic Analysis on the smart contracts and software packages related to the codebase.

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

This review was scoped to the Maya ETH Router repository.

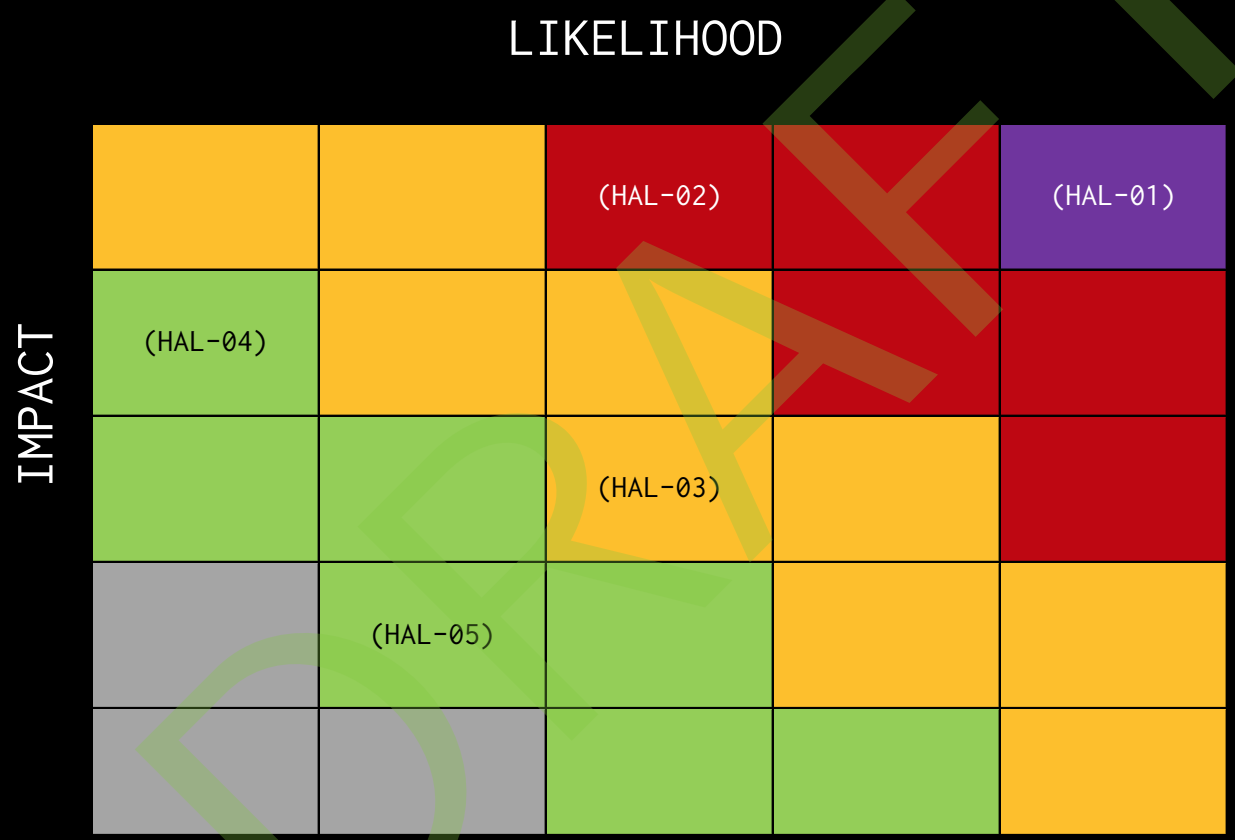
- URL: https://gitlab.com/mayachain/ethereum/eth-router
- Commit hash: 1f48e57bc384169a8ebd9e1ede752eb577b80137

Halborn reviewed the changes made by the Maya team to verify that the modifications are safe.

An in-depth review of the existing THORChain codebase was not in scope for this engagement.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
1	1	1	2	0



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
UNIT TESTS FAILURES INDICATE FUND CALCULATION ISSUES	Critical	SOLVED - 12/09/2022
USE OF TX.ORIGIN CREATES A RISK THAT FUNDS CAN BE STOLEN	High	SOLVED - 12/09/2022
USE OF ERC-20 safeApprove PATTERN CAN BE RISKY	Medium	-
USE OF NPM PACKAGES WITH CRITICAL ISSUES	Low	-
VARIOUS POTENTIAL ISSUES IN UNDERLYING THORCHAIN CODE	Low	-
MULTIPLE IMPLEMENTATIONS OF safeApprove AND safeTransferFrom WITH DIFFERENT BEHAVIOR	Low	-

FINDINGS & TECH DETAILS

3.1 (HAL-01) UNIT TESTS FAILURES INDICATE FUND CALCULATION ISSUES - CRITICAL Critical-

Description:

Running the automated test suite that verifies the functionality of the code results in multiple failures. These tests include coverage of functions that calculate and redistribute users' funds.

Code Location:

The following commands are included in the project's documentation and were executed in order to run the tests:

Listing 1: Commands used to test the ETH router codebase

```
npx hardhat clean
npx hardhat compile
npx hardhat test
```

The project uses hardhat version 2.9.3. Running the above commands using hardhat produced the following output that contain calculation errors.

Listing 2: Output of the command 'npx hardhat test'

```
  Router contract
    User Deposit Assets
       Should Deposit Ether To Asgard1
       Should revert Deposit Ether To Asgard1
     1) Should Deposit RUNE to Asgard1
       Should revert Deposit RUNE to Asgard1
       Should Deposit Token to Asgard1
       Should revert Deposit Token to Asgard1
       Should revert when ETH sent during ERC20 Deposit
       Should Deposit USDT to Asgard1 (38ms)
    Fund Yggdrasil, Yggdrasil Transfer Out
       Should fund yggdrasil ETH
```

```
13        Should fund yggdrasil tokens
14        Should transfer ETH to USER2
15        Should take ETH amount from the amount in transaction,
↳ instead of the amount parameter
16        Should transfer tokens to USER2
17        Should transfer USDT to USER2
18     Yggdrasil Returns Funds, Asgard Churns, Old Vaults can't spend
19        Ygg returns
20        Asgard Churns
21        Should fail to when old Asgard interacts
22        Should fail to when old Yggdrasil interacts
23     Upgrade contract
24        should return vault assets to new router (56ms)
25        should transfer all token and allowance to new contract (75
↳ ms)
26     Evil callbacks
27        should not give more allowance than tokens transfered
28        Test transferOut reverting contract
29        Test transferOut 'to' recipient tries re-entrancy
30        Test re-entrancy protection (generic)
31
32   Aggregator contract
33     Swap In and Out
34        Should Deposit Assets to Router
35       2) Should Swap In Token using Aggregator
36       3) Should Swap In USDT using Aggregator
37       4) Should Swap Out using Aggregator
38       5) Should Fail Swap Out using Aggregator
39       6) Should Fail Swap Out and ETH using Aggregator
40
41
42   24 passing (1s)
43   6 failing
44
45   1) Router contract
46        User Deposit Assets
47          Should Deposit RUNE to Asgard1:
48      Error: Transaction reverted without a reason string
49     at THORChain_Router.safeTransferFrom (contracts/
↳ THORChain_Router.sol:156)
50     at THORChain_Router.deposit (contracts/THORChain_Router.sol
↳ :71)
51     at TruffleContract.deposit (node_modules/@nomiclabs/truffle-
↳ contract/lib/execute.js:169:26)
```

```
52      at Context.<anonymous> (test/1_Router.js:99:30)
53      at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
54
55
56    2) Aggregator contract
57         Swap In and Out
58           Should Swap In Token using Aggregator:
59
60         AssertionError: expected '562099950832084099708' to equal
↳ '562099950491207873777'
61         + expected - actual
62
63         -562099950832084099708
64         +562099950491207873777
65
66         at Context.<anonymous> (test/2_Agg.js:63:60)
67         at runMicrotasks (<anonymous>)
68         at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
69
70    3) Aggregator contract
71         Swap In and Out
72           Should Swap In USDT using Aggregator:
73
74         AssertionError: expected '900000000000000000' to equal
↳ '800000000000000000'
75         + expected - actual
76
77         -900000000000000000
78         +800000000000000000
79
80         at Context.<anonymous> (test/2_Agg.js:93:68)
81         at runMicrotasks (<anonymous>)
82         at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
83
84    4) Aggregator contract
85         Swap In and Out
86           Should Swap Out using Aggregator:
87
88         AssertionError: expected '5620999411776554710857' to equal
↳ '562199940839676347309'
89         + expected - actual
```

```
90
91        -5620999411776554710857
92        +5621999408396763473309
93
94        at Context.<anonymous> (test/2_Agg.js:105:60)
95        at runMicrotasks (<anonymous>)
96        at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
97
98    5) Aggregator contract
99          Swap In and Out
100           Should Fail Swap Out using Aggregator:
101
102       AssertionError: expected '8979972163628467198308' to equal
↳ '8979971348321613598228'
103       + expected - actual
104
105       -8979972163628467198308
106       +8979971348321613598228
107
108       at Context.<anonymous> (test/2_Agg.js:113:59)
109       at runMicrotasks (<anonymous>)
110       at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
111
112    6) Aggregator contract
113          Swap In and Out
114           Should Fail Swap Out and ETH using Aggregator:
115
116       AssertionError: expected '5620999411776554710857' to equal
↳ '5620999333544926678989'
117       + expected - actual
118
119       -5620999411776554710857
120       +5620999333544926678989
121
122       at Context.<anonymous> (test/2_Agg.js:131:60)
123       at runMicrotasks (<anonymous>)
124       at processTicksAndRejections (node:internal/process/
↳ task_queues:96:5)
```

Recommendation:

Review the changes introduced and ensure that they do not affect calculations. If the calculations should, in fact, cause different results in the new version of the code, then the tests should be updated and documented to reflect this.

Consider modifying the CI/CD pipelines in use on the repository such that new code changes cannot be committed to the repository unless all automated tests are passed. This can prevent errors from being introduced during changes to the codebase.

Remediation Plan:

SOLVED: The Maya team has modified the unit tests in a more recent commit, cd6120daecaf5f6b432c250c0668bd4ed5e5a9df.

3.2 (HAL-02) USE OF TX.ORIGIN CREATES A RISK THAT FUNDS CAN BE STOLEN - HIGH

Description:

The contract located in the file contracts/eth_rune.sol makes use of the tx.origin property when transferring funds using the transferTo function.

The use of tx.origin is considered dangerous as it creates a risk for phishing attacks or race conditions in which an attacker is able to steal a user's funds. Transactions using tx.origin will use the authorization and account details of the original sender, an entire set of function calls. Therefore, if a user is tricked into executing a sensitive action via interacting with a malicious contract, the called contract will use the victim's account rather than the malicious contract that actually sent the message.

Code Location:

contracts/eth_rune.sol, Lines 153-162.

Listing 3

```
153   /**
154    * Queries the origin of the tx to enable approval-less
 ↳ transactions, such as for upgrading ETH.RUNE to THOR.RUNE.
155    * Beware phishing contracts that could steal tokens by
 ↳ intercepting tx.origin.
156    * The risks of this are the same as infinite-approved contracts
 ↳  which are widespread.
157    * Acknowledge it is non-standard, but the ERC-20 standard is
 ↳ less-than-desired.
158    */
159   function transferTo(address recipient, uint256 amount) public
 ↳ returns (bool) {
160     _transfer(tx.origin, recipient, amount);
161     return true;
```

```
162    }
```

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

The eth_rune.sol contract is used to support an ERC-20 version of THOR-Chain's RUNE token. The reason for its existence is not relevant to Maya as Maya's chain will token's native its Cosmos blockchain rather than an ERC-20 representation within the Ethreum network.

As Maya has no need to support ERC-20 RUNE, this function can be removed. It may be possible to remove support for ETH-RUNE entirely.

Remediation Plan:

SOLVED: The Maya team has agreed with the assessment that there is no need to support the ERC-20 RUNE on Maya chain. This file has been deleted from the codebase as of commit cd6120daecaf5f6b432c250c0668bd4ed5e5a9df.

3.3 (HAL-03) USE OF ERC-20 safeApprove PATTERN CAN BE RISKY - MEDIUM

Description:

The function safeApprove provides a convenience wrapper function around the function approve that is a part of the ERC-20 function standard.

Although the function is labelled ‘safe’ there still exists a risk in how this function is implemented. If a user calls the approve function more than once without resetting the allowance, the approved user may be able to spend more than intended.

For example, if a user calls approve first with one value X and later change their mind to use the value Y, the user who is approved to spend may be able to in fact spend X + Y, not just Y. This can happen when blocks are ordered in a way such that the approved user issues a spend transaction between the two messages sent by the approving user.

To address this, the EIP-20 standard recommends that the allowance should first be set to 0 before issuing a new approve call. This resets the approved allowance and prevents the approved user from spending more than intended.

Code Location:

There are two separate locations where a wrapper for approve is implemented.

contracts/sushiswap/SushiRouterSmol.sol, Lines 13-17

Listing 4

```
1     function safeApprove(address token, address to, uint value)
↳ internal {
2         // bytes4(keccak256(bytes('approve(address,uint256)')));
```

```
3         (bool success, bytes memory data) = token.call(abi.
↳ encodeWithSelector(0x095ea7b3, to, value));
4         require(success && (data.length == 0 || abi.decode(data, (
↳ bool))), 'TransferHelper: APPROVE_FAILED');
5     }
```

contracts/THORChain_Aggregator.sol, Lines 85-88

Listing 5

```
1     function safeApprove(address _asset, address _address, uint
↳ _amount) internal {
2         (bool success,) = _asset.call(abi.encodeWithSignature("
↳ approve(address,uint256)", _address, _amount)); // Approve to
↳ transfer
3         require(success);
4     }
5 }
```

Recommendation:

It is possible to replace these custom libraries with a project like OpenZeppelin which provides popular, well-tested wrappers for ERC-20 functions. In this case, the safeIncreaseAllowance function should be used instead of safeApprove as the latter is deprecated due to the issues described above.

3.4 (HAL-04) USE OF NPM PACKAGES WITH CRITICAL ISSUES - LOW

Description:

The project uses JavaScript tools such as hardhat in order to build and test Solidity smart contracts. A number of the JavaScript files used contain multiple security issues, including some considered to have critical severity.

We have given this issue a Low risk rating as the vulnerabilities marked with Critical or High severity appear to be relevant primarily in a web context where an attacker can submit malicious values. Maya is using these libraries as local build tools, so it is unlikely that these risks are applicable.

Code Location:

The contents of the JavaScript dependencies can be found in the file package.json.

Listing 6: Contents of package.json

```
{
  "dependencies": {
    "@nomiclabs/hardhat-ethers": "^2.0.2",
    "@nomiclabs/hardhat-truffle5": "^2.0.0",
    "@nomiclabs/hardhat-waffle": "^2.0.1",
    "@nomiclabs/hardhat-web3": "^2.0.0",
    "bignumber.js": "^9.0.0",
    "solc": "^0.7.6",
    "truffle-assertions": "^0.9.2"
  },
  "devDependencies": {
    "chai": "^4.3.4",
    "ethereum-waffle": "^3.3.0",
    "ethers": "^5.1.0",
    "hardhat": "^2.9.3",
    "husky": "^4.2.5",
```

```
17      "web3": "^1.3.4"
18    },
19    "resolutions": {
20      "sha3": "2.0.2"
21    }
22  }
```

The following output was generated using the command npm audit --omit=dev which prints known security vulnerabilities in packages marked as being used in production.

‘‘‘{caption=“npm audit result. Some details have been omitted for clarity”
74 vulnerabilities (9 low, 19 moderate, 20 high, 26 critical)

To address issues that do not require attention, run:
npm audit fix

Listing 7

```
1
2
3 Further details on vulnerable packages can be obtained by running
↳ `npm audit --omit=dev`
4
5 <!--\RiskLevel-->
6
7 ### Recommendation
8
9 As stated above, it is unlikely that these issues expose Maya to
↳ any risk, as the project will not expose these vulnerable
↳ dependencies in a web context.
10
11 At the same time, we recommend removing vulnerabilities in project
↳  dependencies, especially those rating as being High severity or
↳ greater.
12
13 <!--### Remediation Plan-->
14
15 \clearpage
16
17 ## \vuln[2][2]{VARIOUS POTENTIAL ISSUES IN UNDERLYING THORCHAIN
```

```
↳ CODE}
18
19 ### Description
20
21 The ETH router repository is forked from a project created by
↳ THORChain. The codebase is somewhat dated and lacks improvements
↳ introduced in modern versions of Solidity.
22
23 While a full review of the original codebase was outside the scope
↳  of the audit, we have some general recommendations that could
↳ help improve the health of the codebase:
24
25 * Many Solidity files in the codebase use versions of the Solidity
↳  compiler older than version 0.8.0 which introduced 'checked
↳ arithmetic' which prevents issues related to buffer overflow and
↳ underflow. A newer compiler version can be used to access this
↳ security feature.
26
27 * Related to the above, several contracts contain custom "Safe
↳ Math" code that is largely made obsolete by the 0.8.0 release.
↳ After updating, this custom code can be deleted to simplify the
↳ codebase and avoid errors that may exist in custom code.
28
29 * In general, older compiler versions contain known bugs and lack
↳ optimizations introduced in newer releases. Uses more modern
↳ compiler versions will result in safer and more gas-efficient code
↳ .
30
31 \RiskLevel
32
33 ### Code Location
34
35 ```{caption="Examples of outdated Solidity compiler versions"}
36 contracts/EvilToken.sol:pragma solidity 0.7.6;
37 contracts/eth_rune.sol:pragma solidity 0.7.6;
38 contracts/USDT.sol:pragma solidity ^0.4.17;
39 contracts/Token.sol:pragma solidity 0.7.6;
```

Listing 8: Examples of files containing obsolete custom math implementations

```
1 contracts/eth_rune.sol
2 contracts/USDT.sol
```

```
3 contracts/Token.sol
4 contracts/EvilToken.sol
```

Recommendation:

Review the THORChain code to determine if it is possible to make safe updates to the code. This should result in security and performance enhancements.

3.5 (HAL-05) MULTIPLE IMPLEMENTATIONS OF safeApprove AND safeTransferFrom WITH DIFFERENT BEHAVIOR - LOW

Description:

The codebase contains multiple variations of the functions safeApprove and safeTransferFrom. These functions wrap the ERC-20 standard functions approve and transferFrom. This is a common approach taken by many projects in order to reduce usage errors associated with these ERC-20 functions.

The codebase implements safeApprove and safeTransferFrom in two different places, and the functions do not behave in identical ways. As a result, it is possible for errors to occur as a use or developer is likely to assume that two functions with identical names and arguments within the same codebase will exhibit the same behavior.

Code Location:

The following excerpt shows the results of the command rg -S 'function safe*' -A 5 --type solidity which was used to examine the differing functionality of ERC-20 transfer functions.

Listing 9: First implementation of transfer functions

```
contracts/THORChain_Router.sol
153:     function safeTransferFrom(address _asset, uint _amount)
↳ internal returns(uint amount) {
154-         uint _startBal = iERC20(_asset).balanceOf(address(this
↳ ));
155-         (bool success, bytes memory data) = _asset.call(abi.
↳ encodeWithSignature("transferFrom(address,address,uint256)", msg.
↳ sender, address(this), _amount));
156-         require(success && (data.length == 0 || abi.decode(
↳ data, (bool))));
```

```
157-         return (iERC20(_asset).balanceOf(address(this)) - _startBal);
158-     }

contracts/THORChain_Aggregator.sol
78:     function safeTransferFrom(address _asset, uint _amount) internal returns(uint amount) {
79-         uint _startBal = iERC20(_asset).balanceOf(address(this));
80-         (bool success, bytes memory data) = _asset.call(abi.encodeWithSignature("transferFrom(address,address,uint256)", msg.sender, address(this), _amount));
81-         require(success && (data.length == 0 || abi.decode(data, (bool))));
82-         return (iERC20(_asset).balanceOf(address(this)) - _startBal);
83-     }
--
85:     function safeApprove(address _asset, address _address, uint _amount) internal {
86-         (bool success,) = _asset.call(abi.encodeWithSignature("approve(address,uint256)", _address, _amount)); // Approve to transfer
87-         require(success);
88-     }
89-}
```

Listing 10: Second implementation of transfer functions

```
contracts/sushiswap/SushiRouterSmol.sol
13:     function safeApprove(address token, address to, uint value) internal {
14-         // bytes4(keccak256(bytes('approve(address,uint256)')));
15-         (bool success, bytes memory data) = token.call(abi.encodeWithSelector(0x095ea7b3, to, value));
16-         require(success && (data.length == 0 || abi.decode(data, (bool))), 'TransferHelper: APPROVE_FAILED');
17-     }
18-
19:     function safeTransfer(address token, address to, uint value) internal {
20-         // bytes4(keccak256(bytes('transfer(address,uint256)')));
```

```
10 21-        (bool success, bytes memory data) = token.call(abi.
↳ encodeWithSelector(0xa9059cbb, to, value));
11 22-        require(success && (data.length == 0 || abi.decode(data
↳ , (bool))), 'TransferHelper: TRANSFER_FAILED');
12 23-    }
13 24-
14 25:    function safeTransferFrom(address token, address from,
↳ address to, uint value) internal {
15 26-        // bytes4(keccak256(bytes('transferFrom(address,address
↳ ,uint256)')));
16 27-        (bool success, bytes memory data) = token.call(abi.
↳ encodeWithSelector(0x23b872dd, from, to, value));
17 28-        require(success && (data.length == 0 || abi.decode(data
↳ , (bool))), 'TransferHelper: TRANSFER_FROM_FAILED');
18 29-    }
19 30-
20 31:    function safeTransferETH(address to, uint value) internal {
21 32-        (bool success,) = to.call{value:value}(new bytes(0));
22 33-        require(success, 'TransferHelper: ETH_TRANSFER_FAILED')
↳ ;
23 34-    }
24 35-}
25 36-
```

As can be seen above, the safeTransferFrom and safeApprove functions differ in the THORChain and SushiSwap implementations:

- The safeApprove function in particular will revert in different circumstances.
- The implementations in contract/sushiswap/SushiRouterSmol.sol do not return a value.

Recommendation:

It is likely that the THORChain code differs for historical reasons that may be beyond the ability of the project maintainers to modify. However, it may be possible for Maya to use a single implementation of ERC-20 helper functions across its contracts, as they will be deployed in a new context.

Note that there are popular, well-tested implementations of these functions, such as those provided by Solmate and OpenZeppelin. Consider whether it is worth using these libraries rather than rewriting the code shown above.

AUTOMATED TESTING

AUTOMATED TESTING

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped component. Among the tools used were **npm audit** and **slither**. After Halborn verified all the code 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.

Slither - Security Analysis Output Sample:

Severity : error [0]

Severity : warning [53]

- 1-0-locked-ether Contract locking ether found: Contract RevertingContract (_audit/813/RevertingContract.sol#8-12) has payable functions: - RevertingContract.receive() (_audit/813/RevertingContract.sol#9-11) But does not have a function to withdraw the ether : **1**
- 3-0-solc-version Pragma version0.8.13 (_audit/813/RevertingContract.sol#6) allows old versions : **1**
- 0-1-arbitrary-send-eth THORChain_Aggregator.swapIn(address,address,string,address,uint256,uint256,uint256) (_audit/813/THORChain_Aggregator.sol#49-65) sends eth to arbitrary user Dangerous calls: - iROUTER(tcRouter).depositWithExpiry{value: _safeAmount} (address(tcVault),ETH,_safeAmount,tcMemo,deadline) (_audit/813/THORChain_Aggregator.sol#64) : **1**
- 2-1-missing-zero-check THORChain_Aggregator.constructor(address,address)._weth (_audit/813/THORChain_Aggregator.sol#39) lacks a zero-check on : - WETH = _weth (_audit/813/THORChain_Aggregator.sol#41) : **1**
- 3-0-solc-version Pragma version0.8.13 (_audit/813/THORChain_Aggregator.sol#5) allows old versions : **1**
- 3-0-low-level-calls Low level call in THORChain_Aggregator.safeTransferFrom(address,uint256) (_audit/813/THORChain_Aggregator.sol#78-83): - (success,data) = _asset.call(abi.encodeWithSignature(transferFrom(address,address,uint256),msg.sender,address(this),_amount)) (_audit/813/THORChain_Aggregator.sol#80) : **1**
- 3-0-low-level-calls Low level call in THORChain_Aggregator.safeApprove(address,address,uint256) (_audit/813/THORChain_Aggregator.sol#85-88): - (success) = _asset.call(abi.encodeWithSignature(approve(address,uint256),_address,_amount)) (_audit/813/THORChain_Aggregator.sol#86) : **1**

Figure 1: Slither output sample for the ETH Router contracts

No major issue has been found by the Slither.

npm audit - Output:

Listing 11

```
# npm audit report

ajv  <6.12.3
Severity: moderate
Prototype Pollution in Ajv - https://github.com/advisories/GHSA-
↳ v88g-cgmw-v5xw
fix available via `npm audit fix`
node_modules/ajv

ansi-regex  3.0.0
Severity: high
Inefficient Regular Expression Complexity in chalk/ansi-regex -
↳ https://github.com/advisories/GHSA-93q8-gq69-wqmw
fix available via `npm audit fix`
node_modules/ansi-regex

```

```
async  2.0.0 - 2.6.3
Severity: high
Prototype Pollution in async - https://github.com/advisories/GHSA-fwr7-v2mv-hh25
No fix available
node_modules/async
node_modules/ganache-core/node_modules/async
  ganache-core  <=2.1.0-beta.7 || >=2.1.1
  Depends on vulnerable versions of async
  Depends on vulnerable versions of lodash
  Depends on vulnerable versions of web3
  Depends on vulnerable versions of web3-provider-engine
  node_modules/ganache-core
    @ethereum-waffle/provider  <=4.0.1-dev.37f589d || 4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 - 4.0.3-dev.e7e18f6 || 4.0.5-dev.06c4b26 - 4.0.5-dev.edcb2d5
    Depends on vulnerable versions of @ethereum-waffle/ens
    Depends on vulnerable versions of ganache-core
    node_modules/@ethereum-waffle/provider
      @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
      Depends on vulnerable versions of @ethereum-waffle/provider
      node_modules/@ethereum-waffle/chai
        ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-dev.e3fa452
        Depends on vulnerable versions of @ethereum-waffle/chai
        Depends on vulnerable versions of @ethereum-waffle/provider
        node_modules/ethereum-waffle
          @nomiclabs/hardhat-waffle  *
          Depends on vulnerable versions of ethereum-waffle
          node_modules/@nomiclabs/hardhat-waffle

cross-fetch  <=2.2.5 || 3.0.0 - 3.0.5
Severity: moderate
Incorrect Authorization in cross-fetch - https://github.com/advisories/GHSA-7gc6-qh9x-w6h8
Depends on vulnerable versions of node-fetch
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/cross-fetch

css-what  4.0.0 - 5.0.0
Severity: high
Denial of service in css-what - https://github.com/advisories/GHSA-q8pj-2vqx-8ggc
fix available via `npm audit fix`
```

```
node_modules/css-what
  cheerio-select-tmp  *
  Depends on vulnerable versions of css-select
  Depends on vulnerable versions of css-what
  node_modules/cheerio-select-tmp
    cheerio  1.0.0-rc.1 - 1.0.0-rc.5
    Depends on vulnerable versions of cheerio-select-tmp
    node_modules/cheerio
  css-select  3.1.1 - 3.1.2
  Depends on vulnerable versions of css-what
  node_modules/css-select

decode-uri-component  <0.2.1
decode-uri-component vulnerable to Denial of Service (DoS) - https
↳ ://github.com/advisories/GHSA-w573-4hg7-7wgq
fix available via `npm audit fix`
node_modules/decode-uri-component
node_modules/ganache-core/node_modules/decode-uri-component

elliptic  <=6.5.3
Severity: high
Signature Malleabillity in elliptic - https://github.com/
↳ advisories/GHSA-vh7m-p724-62c2
Use of a Broken or Risky Cryptographic Algorithm - https://github.
↳ com/advisories/GHSA-r9p9-mrjm-926w
fix available via `npm audit fix`
node_modules/@nomiclabs/truffle-contract/node_modules/elliptic
node_modules/@truffle/interface-adapter/node_modules/elliptic
node_modules/ganache-core/node_modules/elliptic
  @ethersproject/signing-key  <=5.0.9
  Depends on vulnerable versions of elliptic
  node_modules/ganache-core/node_modules/@ethersproject/signing-
↳ key
  ethers  3.0.0 - 4.0.48 || 5.0.15 - 5.2.0
  Depends on vulnerable versions of @ethersproject/providers
  Depends on vulnerable versions of elliptic
  node_modules/@nomiclabs/truffle-contract/node_modules/ethers
  node_modules/@truffle/interface-adapter/node_modules/ethers
  node_modules/ethers

express  <=4.17.2 || 5.0.0-alpha.1 - 5.0.0-alpha.8
Severity: high
qs vulnerable to Prototype Pollution - https://github.com/
↳ advisories/GHSA-hrpp-h998-j3pp
```

```
Depends on vulnerable versions of body-parser
Depends on vulnerable versions of qs
fix available via `npm audit fix`
node_modules/express
node_modules/ganache-core/node_modules/express

follow-redirects  <=1.14.7
Severity: high
Exposure of Sensitive Information to an Unauthorized Actor in follow-redirects - https://github.com/advisories/GHSA-pw2r-vq6v-hr8c
Exposure of sensitive information in follow-redirects - https://github.com/advisories/GHSA-74fj-2j2h-c42q
fix available via `npm audit fix`
node_modules/follow-redirects

got  <11.8.5
Severity: moderate
Got allows a redirect to a UNIX socket - https://github.com/advisories/GHSA-pfrx-2q88-qq97
No fix available
node_modules/ganache-core/node_modules/got
node_modules/ganache-core/node_modules/swarm-js/node_modules/got
node_modules/got
node_modules/swarm-js/node_modules/got
  swarm-js  0.1.1 - 0.1.17 || 0.1.35 - 0.1.40
  Depends on vulnerable versions of got
  node_modules/ganache-core/node_modules/swarm-js
  node_modules/swarm-js
  web3-bzz  <=1.7.4
  Depends on vulnerable versions of got
  Depends on vulnerable versions of underscore
  node_modules/@truffle/interface-adapter/node_modules/web3-bzz
  node_modules/ganache-core/node_modules/web3-bzz
  node_modules/web3-bzz
    web3  <=1.5.2 || 2.0.0-alpha - 3.0.0-rc.4
    Depends on vulnerable versions of web3-bzz
    Depends on vulnerable versions of web3-core
    Depends on vulnerable versions of web3-eth
    Depends on vulnerable versions of web3-eth-personal
    Depends on vulnerable versions of web3-net
    Depends on vulnerable versions of web3-shh
    Depends on vulnerable versions of web3-utils
    node_modules/@truffle/interface-adapter/node_modules/web3
```

```
132     node_modules/ganache-core/node_modules/web3
133     node_modules/web3
134       @truffle/interface-adapter  <=0.5.6 || >=0.6.0-tezos.0
135       Depends on vulnerable versions of web3
136       node_modules/@truffle/interface-adapter
137         @nomiclabs/truffle-contract  <=4.2.24
138         Depends on vulnerable versions of @truffle/debug-utils
139         Depends on vulnerable versions of @truffle/interface-
 ↳ adapter
140         node_modules/@nomiclabs/truffle-contract
141
142 highlight.js  9.0.0 - 10.4.0
143 Severity: moderate
144 ReDOS vulnerabities: multiple grammars - https://github.com/
 ↳ advisories/GHSA-7wwv-vh3v-89cq
145 fix available via `npm audit fix`
146 node_modules/highlight.js
147   @truffle/debug-utils  1.0.20-alphaTez.0 - 5.0.9
148   Depends on vulnerable versions of @truffle/codec
149   Depends on vulnerable versions of highlight.js
150   node_modules/@truffle/debug-utils
151
152 hosted-git-info  <2.8.9
153 Severity: moderate
154 Regular Expression Denial of Service in hosted-git-info - https://
 ↳ github.com/advisories/GHSA-43f8-2h32-f4cj
155 fix available via `npm audit fix`
156 node_modules/hosted-git-info
157
158 json-schema  <0.4.0
159 Severity: critical
160 json-schema is vulnerable to Prototype Pollution - https://github.
 ↳ com/advisories/GHSA-896r-f27r-55mw
161 fix available via `npm audit fix`
162 node_modules/ganache-core/node_modules/json-schema
163 node_modules/json-schema
164   jsprim  0.3.0 - 1.4.1 || 2.0.0 - 2.0.1
165   Depends on vulnerable versions of json-schema
166   node_modules/ganache-core/node_modules/jsprim
167   node_modules/jsprim
168
169 lodash  <=4.17.20
170 Severity: high
```

```
Prototype Pollution in lodash - https://github.com/advisories/GHSA
 ↳ -p6mc-m468-83gw
Command Injection in lodash - https://github.com/advisories/GHSA
 ↳ -35jh-r3h4-6jhm
Regular Expression Denial of Service (ReDoS) in lodash - https://
 ↳ github.com/advisories/GHSA-29mw-wpgm-hmr9
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/lodash
node_modules/lodash

minimatch  <3.0.5
Severity: high
minimatch ReDoS vulnerability - https://github.com/advisories/GHSA
 ↳ -f8q6-p94x-37v3
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/minimatch
node_modules/minimatch

minimist  <1.2.6
Severity: critical
Prototype Pollution in minimist - https://github.com/advisories/
 ↳ GHSA-xvch-5gv4-984h
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/minimist
node_modules/minimist

node-fetch  <=2.6.6
Severity: high
The `size` option isn't honored after following a redirect in node
 ↳ -fetch - https://github.com/advisories/GHSA-w7rc-rwvf-8q5r
node-fetch is vulnerable to Exposure of Sensitive Information to
 ↳ an Unauthorized Actor - https://github.com/advisories/GHSA-r683-
 ↳ j2x4-v87g
No fix available
node_modules/ganache-core/node_modules/fetch-ponyfill/node_modules
 ↳ /node-fetch
node_modules/ganache-core/node_modules/node-fetch
node_modules/node-fetch
  fetch-ponyfill  1.0.0 - 6.0.2
  Depends on vulnerable versions of node-fetch
  node_modules/ganache-core/node_modules/fetch-ponyfill
    eth-json-rpc-middleware  1.1.0 - 5.0.2
    Depends on vulnerable versions of fetch-ponyfill
    node_modules/ganache-core/node_modules/eth-json-rpc-middleware
```

```
206       eth-json-rpc-infura  <=5.0.0
207       Depends on vulnerable versions of eth-json-rpc-middleware
208       node_modules/ganache-core/node_modules/eth-json-rpc-infura
209         web3-provider-engine  14.0.0 - 15.0.12
210         Depends on vulnerable versions of eth-json-rpc-infura
211         node_modules/ganache-core/node_modules/web3-provider-
 ↳ engine
212
213 normalize-url  4.3.0 - 4.5.0
214 Severity: high
215 ReDoS in normalize-url - https://github.com/advisories/GHSA-px4h-
 ↳ xg32-q955
216 fix available via `npm audit fix`
217 node_modules/ganache-core/node_modules/normalize-url
218 node_modules/normalize-url
219
220 nth-check  <2.0.1
221 Severity: high
222 Inefficient Regular Expression Complexity in nth-check - https://
 ↳ github.com/advisories/GHSA-rp65-9cf3-cjxr
223 fix available via `npm audit fix`
224 node_modules/nth-check
225
226 path-parse  <1.0.7
227 Severity: moderate
228 Regular Expression Denial of Service in path-parse - https://
 ↳ github.com/advisories/GHSA-hj48-42vr-x3v9
229 fix available via `npm audit fix`
230 node_modules/ganache-core/node_modules/path-parse
231 node_modules/path-parse
232
233 qs  6.5.0 - 6.5.2 || 6.7.0 - 6.7.2
234 Severity: high
235 qs vulnerable to Prototype Pollution - https://github.com/
 ↳ advisories/GHSA-hrpp-h998-j3pp
236 qs vulnerable to Prototype Pollution - https://github.com/
 ↳ advisories/GHSA-hrpp-h998-j3pp
237 fix available via `npm audit fix`
238 node_modules/ganache-core/node_modules/body-parser/node_modules/qs
239 node_modules/ganache-core/node_modules/express/node_modules/qs
240 node_modules/ganache-core/node_modules/qs
241 node_modules/qs
242 node_modules/request/node_modules/qs
243   body-parser  1.19.0
```

```
  Depends on vulnerable versions of qs
  node_modules/body-parser
  node_modules/ganache-core/node_modules/body-parser

simple-get  <2.8.2
Severity: high
Exposure of Sensitive Information in simple-get - https://github.
↳ com/advisories/GHSA-wpg7-2c88-r8xv
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/simple-get
node_modules/simple-get

tar  <=4.4.17
Severity: high
Arbitrary File Creation/Overwrite on Windows via insufficient
↳ relative path sanitization - https://github.com/advisories/GHSA
↳ -5955-9wpr-37jh
Arbitrary File Creation/Overwrite via insufficient symlink
↳ protection due to directory cache poisoning using symbolic links -
↳  https://github.com/advisories/GHSA-qq89-hq3f-393p
Arbitrary File Creation/Overwrite via insufficient symlink
↳ protection due to directory cache poisoning using symbolic links -
↳  https://github.com/advisories/GHSA-9r2w-394v-53qc
Arbitrary File Creation/Overwrite due to insufficient absolute
↳ path sanitization - https://github.com/advisories/GHSA-3jfq-g458-7
↳ qm9
Arbitrary File Creation/Overwrite via insufficient symlink
↳ protection due to directory cache poisoning - https://github.com/
↳ advisories/GHSA-r628-mhmh-qjhw
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/tar
node_modules/tar

underscore  1.3.2 - 1.12.0
Severity: critical
Arbitrary Code Execution in underscore - https://github.com/
↳ advisories/GHSA-cf4h-3jhx-xvhq
No fix available
node_modules/@truffle/codec/node_modules/underscore
node_modules/@truffle/interface-adapter/node_modules/underscore
node_modules/ganache-core/node_modules/underscore
node_modules/underscore
node_modules/web3-bzz/node_modules/underscore
node_modules/web3-core-method/node_modules/underscore
```

```
276 node_modules/web3-core-requestmanager/node_modules/underscore
277 node_modules/web3-core-subscriptions/node_modules/underscore
278 node_modules/web3-core/node_modules/underscore
279 node_modules/web3-eth-accounts/node_modules/underscore
280 node_modules/web3-eth-contract/node_modules/underscore
281 node_modules/web3-eth-ens/node_modules/underscore
282 node_modules/web3-eth-personal/node_modules/underscore
283 node_modules/web3-eth/node_modules/underscore
284 node_modules/web3-net/node_modules/underscore
285 node_modules/web3-providers-http/node_modules/underscore
286 node_modules/web3-providers-ipc/node_modules/underscore
287 node_modules/web3-providers-ws/node_modules/underscore
288 node_modules/web3/node_modules/underscore
289   web3-core-helpers  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
290   Depends on vulnerable versions of underscore
291   Depends on vulnerable versions of web3-eth-iban
292   Depends on vulnerable versions of web3-utils
293   node_modules/@truffle/interface-adapter/node_modules/web3-core-
 ↳ helpers
294   node_modules/ganache-core/node_modules/web3-core-helpers
295   node_modules/web3-core-method/node_modules/web3-core-helpers
296   node_modules/web3-core-requestmanager/node_modules/web3-core-
 ↳ helpers
297   node_modules/web3-core-subscriptions/node_modules/web3-core-
 ↳ helpers
298   node_modules/web3-core/node_modules/web3-core-helpers
299   node_modules/web3-eth-accounts/node_modules/web3-core-helpers
300   node_modules/web3-eth-contract/node_modules/web3-core-helpers
301   node_modules/web3-eth-ens/node_modules/web3-core-helpers
302   node_modules/web3-eth-personal/node_modules/web3-core-helpers
303   node_modules/web3-eth/node_modules/web3-core-helpers
304   node_modules/web3-providers-http/node_modules/web3-core-helpers
305   node_modules/web3-providers-ipc/node_modules/web3-core-helpers
306   node_modules/web3-providers-ws/node_modules/web3-core-helpers
307     web3-core-subscriptions  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-
 ↳ rc.4
308     Depends on vulnerable versions of underscore
309     Depends on vulnerable versions of web3-core-helpers
310     node_modules/@truffle/interface-adapter/node_modules/web3-core
 ↳ -subscriptions
311     node_modules/ganache-core/node_modules/web3-core-subscriptions
312     node_modules/web3-core-subscriptions
313       web3-core-method  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
314       Depends on vulnerable versions of underscore
```

```
315       Depends on vulnerable versions of web3-core-helpers
316       Depends on vulnerable versions of web3-core-subscriptions
317       Depends on vulnerable versions of web3-utils
318       node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ core-method
319       node_modules/ganache-core/node_modules/web3-core-method
320       node_modules/web3-core-method
321       web3-shh  <=1.3.5
322       Depends on vulnerable versions of web3-core
323       Depends on vulnerable versions of web3-core-method
324       Depends on vulnerable versions of web3-core-subscriptions
325       Depends on vulnerable versions of web3-net
326       node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ shh
327       node_modules/ganache-core/node_modules/web3-shh
328       node_modules/web3-shh
329     web3-eth-personal  <=1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
330     Depends on vulnerable versions of web3-core
331     Depends on vulnerable versions of web3-core-helpers
332     Depends on vulnerable versions of web3-core-method
333     Depends on vulnerable versions of web3-net
334     Depends on vulnerable versions of web3-utils
335     node_modules/@truffle/interface-adapter/node_modules/web3-eth-
 ↳ personal
336     node_modules/ganache-core/node_modules/web3-eth-personal
337     node_modules/web3-eth-personal
338     web3-providers-http  <=1.0.0 || 1.2.0 - 1.3.5 || 3.0.0-rc.0 -
 ↳ 3.0.0-rc.4
339     Depends on vulnerable versions of web3-core-helpers
340     node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ providers-http
341     node_modules/ganache-core/node_modules/web3-providers-http
342     node_modules/web3-providers-http
343     web3-providers-ipc  <=1.3.6-rc.2 || 3.0.0-rc.0 - 3.0.0-rc.5
344     Depends on vulnerable versions of underscore
345     Depends on vulnerable versions of web3-core-helpers
346     node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ providers-ipc
347     node_modules/ganache-core/node_modules/web3-providers-ipc
348     node_modules/web3-providers-ipc
349     web3-providers-ws  <=1.3.6-rc.2 || 3.0.0-rc.0 - 3.0.0-rc.4
350     Depends on vulnerable versions of underscore
351     Depends on vulnerable versions of web3-core-helpers
```

```
352     node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ providers-ws
353     node_modules/ganache-core/node_modules/web3-providers-ws
354     node_modules/web3-providers-ws
355   web3-core-requestmanager  <=1.3.5 || 3.0.0-rc.0 - 3.0.0-rc.4
356   Depends on vulnerable versions of underscore
357   Depends on vulnerable versions of web3-core-helpers
358   Depends on vulnerable versions of web3-providers-http
359   Depends on vulnerable versions of web3-providers-ipc
360   Depends on vulnerable versions of web3-providers-ws
361   node_modules/@truffle/interface-adapter/node_modules/web3-core-
 ↳ requestmanager
362   node_modules/ganache-core/node_modules/web3-core-requestmanager
363   node_modules/web3-core-requestmanager
364     web3-core  <=1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
365     Depends on vulnerable versions of web3-core-helpers
366     Depends on vulnerable versions of web3-core-method
367     Depends on vulnerable versions of web3-core-requestmanager
368     Depends on vulnerable versions of web3-utils
369     node_modules/@truffle/interface-adapter/node_modules/web3-core
370     node_modules/ganache-core/node_modules/web3-core
371     node_modules/web3-core
372       web3-eth-ens  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
373       Depends on vulnerable versions of underscore
374       Depends on vulnerable versions of web3-core
375       Depends on vulnerable versions of web3-core-helpers
376       Depends on vulnerable versions of web3-eth-abi
377       Depends on vulnerable versions of web3-eth-contract
378       Depends on vulnerable versions of web3-utils
379       node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ eth-ens
380       node_modules/ganache-core/node_modules/web3-eth-ens
381       node_modules/web3-eth-ens
382         web3-eth  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
383         Depends on vulnerable versions of underscore
384         Depends on vulnerable versions of web3-core
385         Depends on vulnerable versions of web3-core-helpers
386         Depends on vulnerable versions of web3-core-method
387         Depends on vulnerable versions of web3-core-subscriptions
388         Depends on vulnerable versions of web3-eth-abi
389         Depends on vulnerable versions of web3-eth-accounts
390         Depends on vulnerable versions of web3-eth-contract
391         Depends on vulnerable versions of web3-eth-ens
392         Depends on vulnerable versions of web3-eth-iban
```

```
393         Depends on vulnerable versions of web3-eth-personal
394         Depends on vulnerable versions of web3-net
395         Depends on vulnerable versions of web3-utils
396         node_modules/@truffle/interface-adapter/node_modules/web3-
 ↳ eth
397         node_modules/ganache-core/node_modules/web3-eth
398         node_modules/web3-eth
399   web3-eth-abi  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
400   Depends on vulnerable versions of underscore
401   Depends on vulnerable versions of web3-utils
402   node_modules/@truffle/interface-adapter/node_modules/web3-eth-
 ↳ abi
403   node_modules/ganache-core/node_modules/web3-eth-abi
404   node_modules/web3-eth-contract/node_modules/web3-eth-abi
405   node_modules/web3-eth-ens/node_modules/web3-eth-abi
406   node_modules/web3-eth/node_modules/web3-eth-abi
407     web3-eth-contract  <=1.3.6-rc.2 || 2.0.0-alpha - 3.0.0-rc.4
408     Depends on vulnerable versions of underscore
409     Depends on vulnerable versions of web3-core
410     Depends on vulnerable versions of web3-core-helpers
411     Depends on vulnerable versions of web3-core-method
412     Depends on vulnerable versions of web3-core-subscriptions
413     Depends on vulnerable versions of web3-eth-abi
414     Depends on vulnerable versions of web3-utils
415     node_modules/@truffle/interface-adapter/node_modules/web3-eth-
 ↳ contract
416     node_modules/ganache-core/node_modules/web3-eth-contract
417     node_modules/web3-eth-contract
418   web3-eth-accounts  <=1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
419   Depends on vulnerable versions of underscore
420   Depends on vulnerable versions of web3-core
421   Depends on vulnerable versions of web3-core-helpers
422   Depends on vulnerable versions of web3-core-method
423   Depends on vulnerable versions of web3-utils
424   node_modules/@truffle/interface-adapter/node_modules/web3-eth-
 ↳ accounts
425   node_modules/ganache-core/node_modules/web3-eth-accounts
426   node_modules/web3-eth-accounts
427   web3-utils  1.0.0-beta.8 - 1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
428   Depends on vulnerable versions of underscore
429   node_modules/@truffle/codec/node_modules/web3-utils
430   node_modules/@truffle/interface-adapter/node_modules/web3-utils
431   node_modules/ganache-core/node_modules/web3-utils
432   node_modules/web3-core-method/node_modules/web3-utils
```

```
433   node_modules/web3-core-requestmanager/node_modules/web3-utils
434   node_modules/web3-core-subscriptions/node_modules/web3-utils
435   node_modules/web3-core/node_modules/web3-utils
436   node_modules/web3-eth-accounts/node_modules/web3-utils
437   node_modules/web3-eth-contract/node_modules/web3-utils
438   node_modules/web3-eth-ens/node_modules/web3-utils
439   node_modules/web3-eth-personal/node_modules/web3-utils
440   node_modules/web3-eth/node_modules/web3-utils
441   node_modules/web3-net/node_modules/web3-utils
442   node_modules/web3-providers-http/node_modules/web3-utils
443   node_modules/web3-providers-ipc/node_modules/web3-utils
444   node_modules/web3-providers-ws/node_modules/web3-utils
445   node_modules/web3/node_modules/web3-utils
446     @truffle/codec  <=0.10.6
447     Depends on vulnerable versions of web3-utils
448     node_modules/@truffle/codec
449     web3-eth-iban  <=1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
450     Depends on vulnerable versions of web3-utils
451     node_modules/@truffle/interface-adapter/node_modules/web3-eth-
 ↳ iban
452     node_modules/ganache-core/node_modules/web3-eth-iban
453     node_modules/web3-core-method/node_modules/web3-eth-iban
454     node_modules/web3-core-requestmanager/node_modules/web3-eth-
 ↳ iban
455     node_modules/web3-core-subscriptions/node_modules/web3-eth-
 ↳ iban
456     node_modules/web3-core/node_modules/web3-eth-iban
457     node_modules/web3-eth-accounts/node_modules/web3-eth-iban
458     node_modules/web3-eth-contract/node_modules/web3-eth-iban
459     node_modules/web3-eth-ens/node_modules/web3-eth-iban
460     node_modules/web3-eth-personal/node_modules/web3-eth-iban
461     node_modules/web3-eth/node_modules/web3-eth-iban
462     node_modules/web3-providers-http/node_modules/web3-eth-iban
463     node_modules/web3-providers-ipc/node_modules/web3-eth-iban
464     node_modules/web3-providers-ws/node_modules/web3-eth-iban
465     web3-net  1.2.0 - 1.3.5 || 2.0.0-alpha - 3.0.0-rc.4
466     Depends on vulnerable versions of web3-core
467     Depends on vulnerable versions of web3-core-method
468     Depends on vulnerable versions of web3-utils
469     node_modules/@truffle/interface-adapter/node_modules/web3-net
470     node_modules/ganache-core/node_modules/web3-net
471     node_modules/web3-net
472
473 undici  <=5.8.1
```

```
Severity: high
ProxyAgent vulnerable to MITM - https://github.com/advisories/GHSA-pgw7-wx7w-2w33
`undici.request` vulnerable to SSRF using absolute URL on `pathname` - https://github.com/advisories/GHSA-8qr4-xgw6-wmr3
Nodejs undici Vulnerable to CRLF Injection via Content-Type - https://github.com/advisories/GHSA-f772-66g8-q5h3
undici before v5.8.0 vulnerable to uncleared cookies on cross-host / cross-origin redirect - https://github.com/advisories/GHSA-q768-x9m6-m9qp
undici before v5.8.0 vulnerable to CRLF injection in request headers - https://github.com/advisories/GHSA-3cvr-822r-rqcc
fix available via `npm audit fix`
node_modules/undici
  hardhat  2.9.0-dev.0 - 2.9.7
  Depends on vulnerable versions of undici
  node_modules/hardhat


ws  5.0.0 - 5.2.2 || 7.0.0 - 7.4.5
Severity: moderate
ReDoS in Sec-Websocket-Protocol header - https://github.com/advisories/GHSA-6fc8-4gx4-v693
ReDoS in Sec-Websocket-Protocol header - https://github.com/advisories/GHSA-6fc8-4gx4-v693
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/web3-provider-engine/node_modules/ws
node_modules/ws
  @ethersproject/providers  <=5.2.0
  Depends on vulnerable versions of ws
  node_modules/@ethersproject/providers

yargs-parser  <=5.0.0
Severity: moderate
yargs-parser Vulnerable to Prototype Pollution - https://github.com/advisories/GHSA-p9pc-299p-vxgp
No fix available
node_modules/@ensdomains/ens/node_modules/yargs-parser
  yargs  4.0.0-alpha1 - 7.0.0-alpha.3 || 7.1.1
  Depends on vulnerable versions of yargs-parser
  node_modules/@ensdomains/ens/node_modules/yargs
    solc  0.3.6 - 0.4.26
    Depends on vulnerable versions of yargs
```

```
508     node_modules/@ensdomains/ens/node_modules/solc
509       @ensdomains/ens  *
510       Depends on vulnerable versions of solc
511       node_modules/@ensdomains/ens
512         @ethereum-waffle/ens  <=4.0.1-dev.e7e18f6 || 4.0.3-dev.06
 ↳ c4b26 - 4.0.3-dev.edcb2d5
513         Depends on vulnerable versions of @ensdomains/ens
514         node_modules/@ethereum-waffle/ens
515
516 74 vulnerabilities (9 low, 19 moderate, 20 high, 26 critical)
517
518 To address issues that do not require attention, run:
519   npm audit fix
520
521 Some issues need review, and may require choosing
522 a different dependency.
```

THANK YOU FOR CHOOSING

// HALBORN