ChainPort

Fix Review

February 2, 2023

Prepared for:
DcentraLab Ltd

Prepared by: **Vasco Franco**

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 100+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
228 Park Ave S #80688
New York, NY 10003
https://www.trailofbits.com
info@trailofbits.com

Notices and Remarks

Copyright and Distribution

Test Coverage Disclaimer

All activities undertaken by Trail of Bits in association with this project were performed in accordance with a statement of work and agreed upon project plan.

Security assessment projects are time-boxed and often reliant on information that may be provided by a client, its affiliates, or its partners. As a result, the findings documented in this report should not be considered a comprehensive list of security issues, flaws, or defects in the target system or codebase.

Trail of Bits uses automated testing techniques to rapidly test the controls and security properties of software. These techniques augment our manual security review work, but each has its limitations: for example, a tool may not generate a random edge case that violates a property or may not fully complete its analysis during the allotted time. Their use is also limited by the time and resource constraints of a project.

When undertaking a fix review, Trail of Bits reviews the fixes implemented for issues identified in the original report. This work involves a review of specific areas of the source code and system configuration, not comprehensive analysis of the system.

Table of Contents

Executive Summary

Engagement Overview

DcentraLab engaged Trail of Bits to review the security of its ChainPort bridge. From May 31 to June 24, 2022, a team of two consultants conducted a security review of the client-provided source code, with eight person-weeks of effort. Details of the project's scope, timeline, test targets, and coverage are provided in the original audit report.

DcentraLab contracted Trail of Bits to review the fixes implemented for issues identified in the original report. On January 30, 2023, one consultant conducted a review of the client-provided source code, with one person-day of effort.

Summary of Findings

The original audit did not uncover any significant flaws or defects that could impact system confidentiality, integrity, or availability. A summary of the original findings is provided below.

EXPOSURE ANALYSIS

Severity	*Count*
High	0
Medium	3
Low	10
Informational	6
Undetermined	3

CATEGORY BREAKDOWN

Category	*Count*
Auditing and Logging	2
Configuration	6
Cryptography	2
Data Exposure	2
Data Validation	3
Denial of Service	1
Patching	1
Testing	1
Timing	1

Undefined Behavior	3

Overview of Fix Review Results

DcentraLab has sufficiently addressed 14 of the 22 issues described in the original audit report.

Project Summary

Contact Information

The following managers were associated with this project:

Dan Guido, Account Manager
dan@trailofbits.com

Mary O'Brien, Project Manager
mary.obrien@trailofbits.com

The following engineers were associated with this project:

Vasco Franco, Consultant
vasco.franco@trailofbits.com

Project Timeline

The significant events and milestones of the project are listed below.

Date	Event
April 28, 2022	Pre-project onboarding architecture call
May 25, 2022	Pre-project kickoff call
June 7, 2022	Status update meeting #1
June 13, 2022	Status update meeting #2
June 21, 2022	Status update meeting #3
June 23, 2022	Delivery of initial report draft
June 23, 2022	Report readout meeting
June 24, 2022	Delivery of second report draft
July 5, 2022	Delivery of final report
September 27, 2022	Review of fixes implemented by DcentraLab
October 6, 2022	Delivery of initial draft of fix review
January 30, 2023	Review of additional fixes implemented by DcentraLab
February 2, 2023	Delivery of final fix review

Project Methodology

Our work in the fix review included the following:

- A review of the findings in the original audit report
- A manual review of the client-provided source code and configuration material

Project Targets

The engagement involved a review of the fixes implemented in the targets listed below.

ChainPort Smart Contracts

Repository	https://github.com/chainport/smart-contracts
Versions	`f10179da0ed2ecaa4e10346023c245f69fd3478e`
	`48ff2a78b3d9aa24b203db308141231c72767152` (Cardano)
Type	Solidity
Platform	EVM

ChainPort Back End

Repository	https://gitlab.com/chainport/chainport-backend
Versions	`a5b218fcd19fb2ecb4e279764c26ac4cf3c2941e`
	`00556e84bfffd965fdb6b91c53a43c8427d0f6d9` (fees)
	`ead0a71784c3a3c488483c55b1e6c5afcf51392a` (Cardano)
Type	Python
Platform	AWS Lambda

ChainPort Front End

Repository	https://gitlab.com/chainport/chainport-app
Version	`32c6ef297ff8168a2aff5ec22c337f15a0951884`
Types	JavaScript, React
Platform	Web

Summary of Fix Review Results

The table below summarizes each of the original findings and indicates whether the issue has been sufficiently resolved. The DcentraLab team has acknowledged the issues that remain unresolved.

ID	Title	Severity	Status
1	Several secrets checked into source control	Medium	Resolved
2	Same credentials used for staging, test, and production environment databases	Low	Resolved
3	Use of error-prone pattern for logging functions	Low	Unresolved
4	Use of hard-coded strings instead of constants	Informational	Unresolved
5	Use of incorrect operator in SQLAlchemy filter	Undetermined	Resolved
6	Several functions receive the wrong number of arguments	Undetermined	Resolved
7	Lack of events for critical operations	Informational	Resolved
8	Lack of zero address checks in setter functions	Informational	Resolved
9	Python type annotations are missing from most functions	Low	Unresolved
10	Use of libraries with known vulnerabilities	Low	Resolved
11	Use of JavaScript instead of TypeScript	Low	Unresolved

12	Use of .format to create SQL queries	Informational	Unresolved
13	Many rules are disabled in the ESLint configuration	Informational	Unresolved
14	Congress can lose quorum after manually setting the quorum value	Medium	Resolved
15	Potential race condition could allow users to bypass PORTX fee payments	Low	Unresolved
16	Signature-related code lacks a proper specification and documentation	Medium	Resolved
17	Cryptographic primitives lack sanity checks and clear function names	Informational	Resolved
18	Use of requests without the timeout argument	Low	Resolved
19	Lack of noopener attribute on external links	Low	Resolved
20	Use of urllib could allow users to leak local files	Undetermined	Unresolved
21	The front end is vulnerable to iFraming	Low	Resolved
22	Lack of CSP header in the ChainPort front end	Low	Resolved

Detailed Fix Review Results

1. Several secrets checked into source control	
Status: **Resolved**	
Severity: **Medium**	Difficulty: **High**
Type: Data Exposure	Finding ID: TOB-CHPT-1
Target: The `chainport-backend` repository	

Description

The `chainport-backend` repository contains several secrets that are checked into source control. Secrets that are stored in source control are accessible to anyone who has had access to the repository (e.g., former employees or attackers who have managed to gain access to the repository).

We used TruffleHog to identify these secrets (by running the command `trufflehog git file://.` in the root directory of the repository). TruffleHog found several types of credentials, including the following, which were verified through TruffleHog's credential verification checks:

- GitHub personal access tokens
- Slack access tokens

TruffleHog also found unverified GitLab authentication tokens and Polygon API credentials.

Furthermore, we found hard-coded credentials, such as database credentials, in the source code, as shown in figure 1.1.

```
[REDACTED]
```

Figure 1.1: `chainport-backend/env.prod.json#L3-L4`

Fix Analysis

The issue is resolved. The DcentraLab team has removed several of the identified secrets from the repository and has added TruffleHog to its CI pipeline.

2. Same credentials used for staging, test, and production environment databases	
Status: **Resolved**	
Severity: **Low**	Difficulty: **High**
Type: Configuration	Finding ID: TOB-CHPT-2
Target: Database authentication	

Description
The staging, test, and production environments' databases have the same username and password credentials.

Fix Analysis
The issue is resolved. The DcentraLab team has updated these credentials to be different for each environment.

3. Use of error-prone pattern for logging functions

Status: **Unresolved**	
Severity: **Low**	Difficulty: **High**
Type: Auditing and Logging	Finding ID: TOB-CHPT-3
Target: The `chainport-backend` repository	

Description

The pattern shown in figure 3.1 is used repeatedly throughout the codebase to log function names.

```
[REDACTED]
```

Figure 3.1: An example of the pattern used by ChainPort to log function names

This pattern is prone to copy-and-paste errors. Developers may copy the code from one function to another but forget to change the function name, as exemplified in figure 3.2.

```
[REDACTED]
```

Figure 3.2: An example of an incorrect use of the pattern used by ChainPort to log function names

We wrote a Semgrep rule to detect these problems (appendix D of the original audit report). This rule detected 46 errors associated with this pattern in the back-end application. Figure 3.3 shows an example of one of these findings.

```
[REDACTED]
```

Figure 3.3: An example of one of the 46 errors resulting from the function-name logging pattern (`chainport-backend/modules/web_3/helpers.py#L313-L315`)

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that it will fix this issue in a future refactor: "(The issue) will be handled in the near future as part of a conventions refactor."

4. Use of hard-coded strings instead of constants

Status: **Unresolved**	
Severity: **Informational**	Difficulty: **High**
Type: Data Validation	Finding ID: TOB-CHPT-4
Target: The `chainport-backend` repository	

Description

The back-end code uses several hard-coded strings that could be defined as constants to prevent any typos from introducing vulnerabilities.

For example, the checks that determine the system's environment compare the result of the `get_env` function with the strings "`develop`", "`staging`", "`prod`", or "`local`". Figure 4.1 shows an example of one of these checks.

```
[REDACTED]
```

Figure 4.1:
chainport-backend/project/lambdas/mainchain/rebalance_monitor.py#L42-L43

We did not find any typos in these literal strings, so we set the severity of this finding to informational. However, the use of hard-coded strings in place of constants is not best practice; we suggest fixing this issue and following other best practices for writing safe code to prevent the introduction of bugs in the future.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that it will fix this issue in a future refactor: "(The issue) will be handled in the near future as part of a conventions refactor."

5. Use of incorrect operator in SQLAlchemy filter

Status: **Resolved**	
Severity: **Undetermined**	Difficulty: **Undetermined**
Type: Undefined Behavior	Finding ID: TOB-CHPT-5
Target: `chainport-backend/project/data/db/port.py#L173`	

Description

The back-end code uses the `is not` operator in an `SQLAlchemy` query's `filter`. `SQLAlchemy` relies on the `__eq__` family of methods to apply the filter; however, the `is` and `is not` operators do not trigger these methods. Therefore, only the comparison operators (`==` or `!=`) should be used.

```
[REDACTED]
```

Figure 5.1: `chainport-backend/project/data/db/port.py#L173`

We did not review whether this flaw could be used to bypass the system's business logic, so we set the severity of this issue to undetermined.

Fix Analysis

The issue is resolved. The operator in the expression has been updated to `!=` in place of `is not`.

6. Several functions receive the wrong number of arguments

Status: **Resolved**	
Severity: **Undetermined**	Difficulty: **Undetermined**
Type: Undefined Behavior	Finding ID: TOB-CHPT-6
Target: The `chainport-backend` repository	

Description

Several functions in the `chainport-backend` repository are called with an incorrect number of arguments:

- Several functions in the `/project/deprecated_files` folder
- A call to `release_tokens_by_maintainer` from the `rebalance_bridge` function (figures 6.1 and 6.2)
- A call to `generate_redeem_signature` from the `regenerate_signature` function (figures 6.3 and 6.4)
- A call to `get_next_nonce_for_public_address` from the `prepare_erc20_transfer_transaction` function (figures 6.5 and 6.6)
- A call to `get_cg_token_address_list` from the main function of the file (likely old debugging code)

```
[REDACTED]
```

Figure 6.1: The `release_tokens_by_maintainer` function is called with four arguments, but at least five are required.
(chainport-backend/project/lambdas/mainchain/rebalance_monitor.py#L109-L114)

```
[REDACTED]
```

Figure 6.2: The definition of the `release_tokens_by_maintainer` function
(chainport-backend/project/lambdas/release_tokens_by_maintainer.py#L27-L34)

```
[REDACTED]
```

Figure 6.3: A call to `generate_redeem_signature` that is missing the `network_id` argument (chainport-backend/project/scripts/keys_maintainers_signature/regenerate_signature.py#L38-L43)

```
[REDACTED]
```

Figure 6.4: The definition of the `generate_redeem_signature` function (chainport-backend/project/lambdas/sidechain/events_handlers/handle_burn_event.py#L46-L48)

```
[REDACTED]
```

Figure 6.5: A call to `get_next_nonce_for_public_address` that is missing the `outer_session` argument (chainport-backend/project/web3_cp/erc20/prepare_erc20_transfer_transaction.py#L32-L34)

```
[REDACTED]
```

Figure 6.6: The definition of the `get_next_nonce_for_public_address` function (chainport-backend/project/web3_cp/nonce.py#L19-L21)

```
[REDACTED]
```

Figure 6.7: A call to `get_cg_token_address_list` that is missing all three arguments (chainport-backend/project/lambdas/token_endpoints/cg_list_get.py#L90-91)

```
[REDACTED]
```

Figure 6.8: The definition of the `get_cg_token_address_list` function (chainport-backend/project/lambdas/token_endpoints/cg_list_get.py#L37)

We did not review whether this flaw could be used to bypass the system's business logic, so we set the severity of this issue to undetermined.

Fix Analysis

The issue is resolved. The affected code has been removed or updated to pass the correct number of arguments.

7. Lack of events for critical operations	
Status: **Resolved**	
Severity: **Informational**	Difficulty: **High**
Type: Auditing and Logging	Finding ID: TOB-CHPT-7
Target: `ChainportMainBridge.sol`, `ChainportSideBridge.sol`, `Validator.sol`	

Description
Several critical operations do not trigger events. As a result, it will be difficult to review the correct behavior of the contracts once they have been deployed.

For example, the `setSignatoryAddress` function, which is called in the `Validator` contract to set the signatory address, does not emit an event providing confirmation of that operation to the contract's caller (figure 7.1).

```
[REDACTED]
```

Figure 7.1: The `setSignatoryAddress` function in `Validator:43-52`

Without events, users and blockchain-monitoring systems cannot easily detect suspicious behavior.

Fix Analysis
The issue is resolved. The DcentraLab team implemented event triggers on all critical operations.

8. Lack of zero address checks in setter functions

Status: **Resolved**	
Severity: **Informational**	Difficulty: **High**
Type: Data Validation	Finding ID: TOB-CHPT-8
Target: `ChainportMainBridge.sol`, `ChainportMiddleware.sol`, `ChainportSideBridge.sol`	

Description
Certain setter functions fail to validate incoming arguments, so callers can accidentally set important state variables to the zero address.

For example, in the `initialize` function of the `ChainportMainBridge` contract, developers can define the maintainer registry, the congress address for governance, and the signature validator and set their addresses to the zero address.

```
[REDACTED]
```

Figure 8.1: The `initialize` function of `ChainportMainBridge.sol`

Failure to immediately reset an address that has been set to the zero address could result in unexpected behavior.

Fix Analysis
The issue is resolved. The DcentraLab team now runs a script that verifies that important state variables are not set to zero before deploying the contracts.

9. Python type annotations are missing from most functions

Status: **Unresolved**	
Severity: **Low**	Difficulty: **High**
Type: Undefined Behavior	Finding ID: TOB-CHPT-9
Target: The `chainport-backend` repository	

Description
The back-end code uses Python type annotations; however, their use is sporadic, and most functions are missing them.

Fix Analysis
The issue has not been resolved. The DcentraLab team noted that it will fix this issue in a future refactor: "(The issue) will be handled in the near future as part of a conventions refactor."

10. Use of libraries with known vulnerabilities	
Status: **Resolved**	
Severity: **Low**	Difficulty: **Low**
Type: Patching	Finding ID: TOB-CHPT-10
Target: The `chainport-backend` repository	

Description

The back-end repository uses outdated libraries with known vulnerabilities. We used `pip-audit`, a tool developed by Trail of Bits with support from Google to audit Python environments and dependency trees for known vulnerabilities, and identified two known vulnerabilities in the project's dependencies (as shown in figure 10.1).

```
[REDACTED]
```

Figure 10.1: A list of outdated libraries in the back-end repository

Fix Analysis

The issue is resolved. The DcentraLab team updated the back end's dependencies. However, we still recommend integrating `pip-audit` into the CI/CD pipeline to have continuous scanning for Python packages with known vulnerabilities.

11. Use of JavaScript instead of TypeScript

Status: **Unresolved**	
Severity: **Low**	Difficulty: **Low**
Type: Configuration	Finding ID: TOB-CHPT-11
Target: The `chainport-app` repository	

Description

The ChainPort front end is developed with JavaScript instead of TypeScript. TypeScript is a strongly typed language that compiles to JavaScript. It allows developers to specify the types of variables and function arguments, and TypeScript code will fail to compile if there are type mismatches. Contrarily, JavaScript code will crash (or worse) during runtime if there are type mismatches.

In summary, TypeScript is preferred over JavaScript for the following reasons:

- It improves code readability; developers can easily identify variable types and the types that functions receive.
- It improves security by providing static type checking that catches errors during compilation.
- It improves support for integrated development environments (IDEs) and other tools by allowing them to reason about the types of variables.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that the move from JavaScript to TypeScript will occur in a front-end refactor.

12. Use of .format to create SQL queries

Status: **Unresolved**	
Severity: **Informational**	Difficulty: **Medium**
Type: Data Validation	Finding ID: TOB-CHPT-12
Target: [REDACTED]	

Description

The back end builds SQL queries with the `.format` function. An attacker that controls one of the variables that the function is formatting will be able to inject SQL code to steal information or damage the database.

```
[REDACTED]
```

Figure 12.1: chainport-backend/project/data/db/postgres.py#L4-L24

```
[REDACTED]
```

Figure 12.2: chainport-backend/project/lambdas/database_monitor/clear_lock.py#L29-L31

None of the fields described above are attacker-controlled, so we set the severity of this finding to informational. However, the use of `.format` to create SQL queries is an anti-pattern; parameterized queries should be used instead.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that it will fix this issue in a future refactor: "(The issue) will be handled in the near future as part of a conventions refactor."

13. Many rules are disabled in the ESLint configuration

Status: **Unresolved**	
Severity: **Informational**	Difficulty: **High**
Type: Testing	Finding ID: TOB-CHPT-13
Target: `chainport-app/.eslintrc.js`	

Description

There are 34 rules disabled in the front-end `eslint` configuration. Disabling some of these rules does not cause problems, but disabling others reduces the code's security and reliability (e.g., `react/no-unescaped-entities`, `consistent-return`, `no-shadow`) and the code's readability (e.g., `react/jsx-boolean-value`, `react/jsx-one-expression-per-line`).

Furthermore, the code contains 46 inline `eslint-disable` comments to disable specific rules. While disabling some of these rules in this way may be valid, we recommend adding a comment to each instance explaining why the specific rule was disabled.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that this issue will be resolved in a front-end refactor.

14. Congress can lose quorum after manually setting the quorum value

Status: **Resolved**	
Severity: **Medium**	Difficulty: **High**
Type: Configuration	Finding ID: TOB-CHPT-14
Target: `contracts/governance/ChainportCongressMembersRegistry.sol`	

Description

Proposals to the ChainPort congress must be approved by a minimum quorum of members before they can be executed. By default, when a new member is added to the congress, the quorum is updated to be N − 1, where N is the number of congress members.

```
[REDACTED]
```

Figure 14.1: smart-contracts/contracts/governance/ChainportCongressMembersRegistry.sol#L98-L119

However, the congress has the ability to overwrite the quorum number to any nonzero number, including values larger than the current membership.

```
[REDACTED]
```

Figure 14.2: smart-contracts//contracts/governance/ChainportCongressMembersRegistry.sol#L69-L77

If the congress manually lowers the quorum number and later adds a member, the quorum number will be reset to one less than the total membership. If for some reason certain members are temporarily or permanently unavailable (e.g., they are on vacation or their private keys were destroyed), the minimum quorum would not be reached.

Fix Analysis

The issue is resolved. The DcentraLab team indicated that this is desired behavior and documented it in the project's README.

15. Potential race condition could allow users to bypass PORTX fee payments

Status: **Unresolved**	
Severity: **Low**	Difficulty: **Medium**
Type: Timing	Finding ID: TOB-CHPT-15
Target: `contracts/ChainportFeeManager.sol`	

Description

ChainPort fees are paid either as a 0.3% fee deducted from the amount transferred or as a 0.2% fee in PORTX tokens that the user has deposited into the `ChainportFeeManager` contract. To determine whether a fee should be paid in the base token or in PORTX, the back end checks whether the user has a sufficient PORTX balance in the `ChainportFeeManager` contract.

```
[REDACTED]
```

Figure 15.1: chainport-backend//project/lambdas/fees/fees.py#L219-249

However, the `ChainportFeeManager` contract does not enforce an unbonding period, a period of time before users can unstake their PORTX tokens.

```
[REDACTED]
```

Figure 15.2: smart-contracts/contracts/ChainportFeeManager.sol#L113-L125

Since pending fee payments are generated as part of deposit, transfer, and burn events but the actual processing is handled by a separate monitor, it could be possible for a user to withdraw her PORTX tokens on-chain after the deposit event has been processed and before the fee payment transaction is confirmed, allowing her to avoid paying a fee for the transfer.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that it is currently working on a fix.

16. Signature-related code lacks a proper specification and documentation	
Status: **Resolved**	
Severity: **Medium**	Difficulty: **High**
Type: Cryptography	Finding ID: TOB-CHPT-16
Target: Signature-related code	

Description

ChainPort uses signatures to ensure that messages to mint and release tokens were generated by the back end. These signatures are not well documented, and the properties they attempt to provide are often unclear. For example, answers to the following questions are not obvious; we provide example answers that could be provided in the documentation of ChainPort's use of signatures:

- **Why does the signed message contain a `networkId` field, and why does it have to be unique?** If not, an operation to mint tokens on one chain could be replayed on another chain.
- **Why does the signed message contain an `action` field?** The `action` field prevents replay attacks in networks that have both a main and side bridge. Without this field, a signature for minting tokens could be used on a sidechain contract of the same network to release tokens.
- **Why are both the signature and nonce checked for uniqueness in the contracts?** The signatures could be represented in more than one format, which means that storing them is not enough to ensure uniqueness.

Fix Analysis

The issue is resolved. The DcentraLab team added code comments to the relevant source file to document answers to the questions listed above.

17. Cryptographic primitives lack sanity checks and clear function names

Status: **Resolved**	
Severity: **Informational**	Difficulty: **High**
Type: Cryptography	Finding ID: TOB-CHPT-17
Target: `chainport-backend/modules/cryptography_2key/signatures.py`	

Description

Several cryptographic primitives are missing sanity checks on their inputs. Without such checks, problems could occur if the primitives are used incorrectly.

The `remove_0x` function (figure 17.1) does not check that the input starts with `0x`. A similar function in the `eth-utils` library has a more robust implementation, as it includes a check on its input (figure 17.2).

```
[REDACTED]
```

Figure 17.1: chainport-backend/modules/cryptography_2key/signatures.py#L10-L16

```
[REDACTED]
```

Figure 17.2: ethereum/eth-utils/eth_utils/hexadecimal.py#L43-L46

The `add_leading_0` function's name does not indicate that the value is padded to a length of 64 (figure 17.3).

```
[REDACTED]
```

Figure 17.3: chainport-backend/modules/cryptography_2key/signatures.py#L19-L25

The `_build_withdraw_message` function does not ensure that the `beneficiary_address` and `token_address` inputs have the expected length of 66 bytes and that they start with `0x` (figure 17.4).

```
[REDACTED]
```

Figure 17.4: chainport-backend/modules/cryptography_2key/signatures.py#L28-62

We did not identify problems in the way these primitives are currently used in the code, so we set the severity of this finding to informational. However, if the primitives are used improperly in the future, cryptographic bugs that can have severe consequences could be introduced, which is why we highly recommend fixing the issues described in this finding.

Fix Analysis

The issue is resolved. The code now uses the `remove_0x_prefix` function from the `eth-utils` library.

18. Use of requests without the timeout argument	
Status: **Resolved**	
Severity: **Low**	Difficulty: **High**
Type: Denial of Service	Finding ID: TOB-CHPT-18
Target: The `chainport-backend` repository	

Description

The Python `requests` library is used in the ChainPort back end without the `timeout` argument. By default, the `requests` library will wait until the connection is closed before fulfilling a request. Without the `timeout` argument, the program will hang indefinitely.

The following locations in the back-end code are missing the timeout argument:

- `chainport-backend/modules/coingecko/api.py#L29`
- `chainport-backend/modules/requests_2key/requests.py#L14`
- `chainport-backend/project/stats/cg_prices.py#L74`
- `chainport-backend/project/stats/cg_prices.py#L95`

The code in these locations makes requests to the following websites:

- https://api.coingecko.com
- https://ethgasstation.info
- https://gasstation-mainnet.matic.network

If any of these websites hang indefinitely, so will the back-end code.

Fix Analysis

The issue is resolved. Every request made by the ChainPort back end now uses a `timeout` argument.

19. Lack of noopener attribute on external links

Status: **Resolved**	
Severity: **Low**	Difficulty: **High**
Type: Configuration	Finding ID: TOB-CHPT-19
Target: `chainport-app/src/modules/exchange/components/PortOutModal.jsx`	

Description

In the ChainPort front-end application, there are links to external websites that have the `target` attribute set to `_blank` but lack the `noopener` attribute. Without this attribute, an attacker could perform a reverse tabnabbing attack.

```
[REDACTED]
```

Figure 19.1: chainport-app/src/modules/exchange/components/PortOutModal.jsx#L126

Fix Analysis

The issue is resolved. Every link to an external website now includes the `rel="noopener noreferrer"` attribute.

20. Use of urllib could allow users to leak local files

Status: **Unresolved**	
Severity: **Undetermined**	Difficulty: **High**
Type: Data Exposure	Finding ID: TOB-CHPT-20
Target: `chainport-backend/modules/infrastructure/aws/s3.py`	

Description

To upload images of new tokens to S3, the `upload_media_from_url_to_s3` function uses the `urllib` library (figure 20.1), which supports the `file://` scheme; therefore, if a malicious actor controls a dynamic value uploaded to S3, she could read arbitrary local files.

```
[REDACTED]
```

Figure 20.1: `chainport-backend/modules/infrastructure/aws/s3.py#L25-29`

The code in figure 20.2 replicates this issue.

```
[REDACTED]
```

Figure 20.2: Code to test `urlopen`'s support of the `file://` scheme

We set the severity of this finding to undetermined because it is unclear whether an attacker (e.g., a token owner) would have control over token images uploaded to S3 and whether the server holds files that an attacker would want to extract.

Fix Analysis

The issue has not been resolved. The DcentraLab team noted that it will fix this issue in a future refactor: "(The issue) will be handled in the near future as part of a conventions refactor."

21. The front end is vulnerable to iFraming

Status: **Resolved**	
Severity: **Low**	Difficulty: **High**
Type: Configuration	Finding ID: TOB-CHPT-21
Target: The `chainport-app` repository	

Description
The ChainPort front end does not prevent other websites from iFraming it.

Figure 21.1 shows an example of how another website could iFrame the ChainPort front end.

```
[REDACTED]
```

Figure 21.1: An example of how another website could iFrame the ChainPort front end

Fix Analysis
The issue is resolved. The server now replies with `X-Frame-Options: DENY` to prevent other websites from iFraming it. However, the DcentraLab team also added a CSP to fix the issue described in TOB-CHPT-22; the added CSP means that this `X-Frame-Options` option is redundant and can safely be removed. The CSP's `frame-src` does the same job.

22. Lack of CSP header in the ChainPort front end

Status: **Resolved**	
Severity: **Low**	Difficulty: **High**
Type: Configuration	Finding ID: TOB-CHPT-22
Target: The `chainport-app` repository	

Description

The ChainPort front end lacks a Content Security Policy (CSP) header, leaving it vulnerable to cross-site scripting (XSS) attacks.

A CSP header adds extra protection against XSS and data injection attacks by enabling developers to select the sources that the browser can execute or render code from. This safeguard requires the use of the CSP HTTP header and appropriate directives in every server response.

Fix Analysis

The issue is resolved. A CSP header has been added to the ChainPort application.

A. Status Categories

The following table describes the statuses used to indicate whether an issue has been sufficiently addressed.

Fix Status	
Status	**Description**
Undetermined	The status of the issue was not determined during this engagement.
Unresolved	The issue persists and has not been resolved.
Partially Resolved	The issue persists but has been partially resolved.
Resolved	The issue has been sufficiently resolved.

B. Vulnerability Categories

The following tables describe the vulnerability categories, severity levels, and difficulty levels used in this document.

Vulnerability Categories	
Category	**Description**
Access Controls	Insufficient authorization or assessment of rights
Auditing and Logging	Insufficient auditing of actions or logging of problems
Authentication	Improper identification of users
Configuration	Misconfigured servers, devices, or software components
Cryptography	A breach of system confidentiality or integrity
Data Exposure	Exposure of sensitive information
Data Validation	Improper reliance on the structure or values of data
Denial of Service	A system failure with an availability impact
Error Reporting	Insecure or insufficient reporting of error conditions
Patching	Use of an outdated software package or library
Session Management	Improper identification of authenticated users
Testing	Insufficient test methodology or test coverage
Timing	Race conditions or other order-of-operations flaws
Undefined Behavior	Undefined behavior triggered within the system

Severity Levels	
Severity	**Description**
Informational	The issue does not pose an immediate risk but is relevant to security best practices.
Undetermined	The extent of the risk was not determined during this engagement.
Low	The risk is small or is not one the client has indicated is important.
Medium	User information is at risk; exploitation could pose reputational, legal, or moderate financial risks.
High	The flaw could affect numerous users and have serious reputational, legal, or financial implications.

Difficulty Levels	
Difficulty	**Description**
Undetermined	The difficulty of exploitation was not determined during this engagement.
Low	The flaw is well known; public tools for its exploitation exist or can be scripted.
Medium	An attacker must write an exploit or will need in-depth knowledge of the system.
High	An attacker must have privileged access to the system, may need to know complex technical details, or must discover other weaknesses to exploit this issue.