Quantstamp Security Assessment Certificate

May 6th 2022 — Quantstamp Verified

LI.FI

This smart contract audit was prepared by Quantstamp, the protocol for securing smart contracts.

Executive Summary

Type	Cross-Chain Applications Aggregator
Auditors	Poming Lee, Senior Research Engineer Rabib Islam, Research Engineer Guillermo Escobero, Security Auditor
Timeline	2022-04-11 through 2022-05-05
EVM	Arrow Glacier
Languages	Solidity
Methods	Architecture Review, Computer-Aided Verification, Manual Review
Specification	README.md official-documentation
Documentation Quality	Medium
Test Quality	Medium

Source Code

Repository	Commit
lifinance/contracts	7a4a667
lifinance/contracts	1e6e555

Total Issues	**10** (6 Resolved)
High Risk Issues	**1** (1 Resolved)
Medium Risk Issues	**1** (1 Resolved)
Low Risk Issues	**2** (1 Resolved)
Informational Risk Issues	**6** (3 Resolved)
Undetermined Risk Issues	0 (0 Resolved)

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.
Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Fixed	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

During the audit, we found 10 potential issues of various levels of severity: 1 high-severity issues, 1 medium-severity issues, 2 low-severity issues, and 6 informational-severity issues. We also made 6 best practices recommendations. We highly recommend addressing all the comments and findings before going live.

2022-05-05 update: During this re-audit, the admin team has brought all the status of findings into fixed or acknowledged.

ID	Description	Severity	Status
QSP-1	Potential Unexpected Collision Caused by Unmapped Address of `LibStorage`s	⇈ High	Mitigated
QSP-2	`jlength` Not Updated After a DEX is Removed From the Storage	^ Medium	Fixed
QSP-3	Diamond Facet Upgrade Could Lead to Inconsistency	⌄ Low	Acknowledged
QSP-4	Events Should Be Emitted In Critical Operations	⌄ Low	Fixed
QSP-5	Highly Dependent On Offchain Components	○ Informational	Acknowledged
QSP-6	Clone-And-Own	○ Informational	Fixed
QSP-7	Two Versions Of `Swapper.sol`	○ Informational	Fixed
QSP-8	Missing Input Check	○ Informational	Acknowledged
QSP-9	Privileged Roles and Ownership	○ Informational	Acknowledged
QSP-10	Unlocked Pragma	○ Informational	Fixed

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

DISCLAIMER: The scope of the current audit is for all contract files ` /src` except ` /src/Libraries/LibBytes.sol` .

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.8.3

Steps taken to run the tools:

1. Installed the Slither tool: `pip install slither-analyzer`
2. Run Slither from the project directory: `slither .`

Findings

QSP-1 Potential Unexpected Collision Caused by Unmapped Address of `LibStorage`s

Severity: *High Risk*

Status: Mitigated

File(s) affected: `src/Facets/DexManagerFacet.sol`, `src/Helpers/Swapper.sol`

Description: There is no predefined addresses for:

- `newOwner` in `src/Facets/OwnershipFacet.sol`
- `LibStorage s` in `src/Facets/DexManagerFacet.sol`
- `LibStorage ls` in `src/Facets/Swapper.sol`. This collision on slot `0` needs to be resolved.

Recommendation: It is suggested to specify predefined random memory addresses for those storages, like other facets (e.g., on `L20` in `src/Facets/HopFacet.sol`).

Update: The current collision in the code (i.e., the `newOwner` part) is resolved. For the rest of the storages, the admin team stated that "LibStorage is always at slot 0 and meant to be accessed globally by all facets. All local storage will continue to use namespaced storage slots". Quantstamp acknowledged that. However, it is worth to note that this design might accidentally lead to possible collisions in the future when adding new facets.

QSP-2 `jlength` Not Updated After a DEX is Removed From the Storage

Severity: *Medium Risk*

Status: Fixed

Description: On `L95` in `src/Facets/DexManagerFacet.sol`, the `jlength` is never updated after a DEX is removed from the storage using `_removeDex(j)`.

Recommendation: Consider updating the `jlength` whenever the `storageDexes.length` has changed.

Update: The admin team has fixed this issue based on the recommendation.

QSP-3 Diamond Facet Upgrade Could Lead to Inconsistency

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `src/Libraries/LibDiamond.sol`, `src/Facets/DiamondCutFacet.sol`

Description: If during an upgrade `diamondCut(...)` calls are executed in multiple Ethereum transactions, users may be exposed to contracts that are upgraded only partially (i.e., some of the functions are upgraded while others are not). This may result in unexpected inconsistencies.

Recommendation: We recommend upgrading the contracts in a single transaction, or making the fallback function pausable for the duration of an upgrade.

Update: The admin team stated that "Upgrades of new facets are done in a single transaction already."

QSP-4 Events Should Be Emitted In Critical Operations

Severity: *Low Risk*

Status: Fixed

File(s) affected: `src/Facets/HopFacet.sol`, `src/Facets/HyphenFacet.sol`, `src/Facets/NXTPFacet.sol`

Description: Several contracts implement an "initializer" function. These functions can be called many times, modifying the storage of the contract. This includes critical variables, such as addresses pointing to tokens and bridges.
The execution of these functions should emit events to track misuse. `CBridgeFacet` emits `Inited` event (declared in `ILiFi` interface) when initialized. However, `HopFacet`, `HyphenFacet` and `NXTPFacet` contracts do not emit events when initialized or when critical changes are made.

Recommendation: Emit events every time `HopFacet`, `HyphenFacet`, and `NXTPFacet` are initialized.

Update: The admin team has fixed this issue based on the recommendation.

QSP-5 Highly Dependent On Offchain Components

Severity: *Informational*

Status: Acknowledged

File(s) affected: `All the facets for bridging.`

Description: For all the bridging facets the input parameters of its functions that are open to arbitrary end users are all generated by offchain components which can be hacked or DDoSed and the data passed in are never checked.

Recommendation: This is the nature of the current design and it is understandable. The admin team would have to do their best on making sure that the offchain components are correctly functioning 24/7/365.

Update: The admin team stated that "The contract is designed to use off-chain data. It is also open for anyone to use independently of the LI.FI web app so there is no hard dependency".

QSP-6 Clone-And-Own

Severity: *Informational*

Status: Fixed

File(s) affected: `src/Facets/DiamondCutFacet.sol`, `src/Facets/DiamondLoupeFacet.sol`, `src/Libraries/LibDiamond.sol`

Description: LiFinance project uses the EIP-2535 standard implementation. In the related documentation (`docs/LibDiamond.md`) is stated that is copied from the proposed implementation of the EIP author: `mudgen/diamond-1-hardhat` GitHub repository. However, we found that the files have minor differences between them.

Recommendation: If the code is cloned from another repository, it is a good practice to add comments to the cloned files, including the commit hash of the file and clarifying any modifications done from that point. This can help to improve the traceability and maintainability of the project.

Update: The admin team has fixed this issue based on the recommendation.

QSP-7 Two Versions Of `Swapper.sol`

Severity: *Informational*

Status: Fixed

File(s) affected: `src/Facets/Swapper.sol`, `src/Helpers/Swapper.sol`

Description: We noticed that there are two files named `Swapper.sol` with similar functionality. Observing the git history, we think that `src/Facets/Swapper.sol` was not removed when

refactored into `src/Helpers/Swapper.sol`. Also, no contract imports or inherits from `src/Facets/Swapper.sol` (i.e., this file is never used). This can lead to confusion and unexpected behavior if a contract imports the wrong file.

Recommendation: Delete the wrong file or clarify the project files structure.

Update: The admin team has fixed this issue based on the recommendation.

QSP-8 Missing Input Check

Severity: Informational

Status: Acknowledged

File(s) affected: `src/LiFiDiamond.sol`

Description: `constructor` in `src/LiFiDiamond.sol` does not check if `address _contractOwner` and `address _diamondCutFacet` are non-zero addresses.

Recommendation: Consider adding zero address checks to the `constructor`.

Update: The admin team stated that "Deployment is done from a deploy script no more than once. No need to add extra checks as the system would not even work if the owner set to 0 on deploy anyway".

QSP-9 Privileged Roles and Ownership

Severity: Informational

Status: Acknowledged

File(s) affected: `All contracts`

Description: There are some actions that could have important consequences for end-users.

1. The programs deployed are upgradable and can be changed by the admin team at any time at will.
2. Based on `L22` of `src/Facets/WithdrawFacet.sol`, the admin team of the platform can transfer whatever ether/ERC20 assets that are held by the platform to any address for rescue purposes.
3. Based on `src/Facets/DexManagerFacet.sol`, the platform owner can approve any dex and any function signature.

Recommendation: The centralization of power needs to be made clear to the users, especially depending on the level of privilege the contract allows to the owner.

Update: The admin team has added this information to their public facing documents.

QSP-10 Unlocked Pragma

Severity: Informational

Status: Fixed

File(s) affected: `All contracts`

Description: Every Solidity file specifies in the header a version number of the format pragma solidity (^)0.8.*. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version and above, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, it is recommended to remove the caret to lock the file onto a specific Solidity version.

Update: The admin team has fixed this issue based on the recommendation.

Automated Analyses

Slither

Slither reported `220` results, all of which were either identified as false positives or included in the findings of this report.

Code Documentation

1. (update: fixed) Code is missing NatSpec for some functions and should include more inline comments describing variables.
2. (update: fixed) In some cases, section headers in the code do not match with their contents (e.g. `_executeAndCheckSwaps(...)` in `Swapper.sol` should be under "Private Methods" if it is to be grouped with `_executeSwaps(...)`).
3. (update: fixed) Error in `AnyswapFacet.md`:
   . The "Hop"Facet works by forwarding Anyswap... should be:
   . The "Anyswap"Facet works by forwarding Anyswap...
4. (update: fixed) Comment in `L13` of `GenericSwapFacet.sol` can be confusing: "@notice Provides functionality for swapping through ANY DEX".
   . DEXs used for swapping have to be approved first using `DexManagerFacet`.
   . Consider modifying it to: "@notice Provides functionality for swapping through ANY APPROVED DEX".

Adherence to Best Practices

1. (update: fixed) Variable naming should be more descriptive for the `LibStorage` variable in all contracts using it.
2. (update: fixed) `_bridge()` function in `CBridgeFacet.sol` is not used. Consider removing it.

3. (update: fixed) `error InvalidConfig()` is declared several times throughout the codebase. Consider declaring it in `GenericErrors.sol`, a file that contains the generic errors used in the project.
4. (update: fixed) `keccak256("com.lifi.facets.hyphen")` in `HyphenFacet.sol` can be computed offchain and initialized using a hex value, as done in the other contracts.
5. (update: fixed) `getStorage()` in `HyphenFacet` is declared as `internal`. In other contracts that use the same pattern, it is declared as `private`. Consider modifying it to `private` to keep consistency.
6. (update: fixed) `_executeSwaps` in `Swapper.sol` is expected to be `private`, but it is declared as `internal`. Consider modifying it to `private` to keep consistency.

Test Results

Test Suite Results

All tests has passed.

```
==yarn test

yarn run v1.22.15
$ npm run compile && cross-env TS_NODE_TRANSPILE_ONLY=1 mocha --bail --recursive test
npm WARN lifecycle The node binary used for scripts is /tmp/yarn--1651499161483-0.94958962529306/node but npm is using /opt/hostedtoolcache/node/14.19.1/x64/bin/node itself. Use the `--scripts-prepend-node-path` option to
include the path for the node binary npm was executed with.

> lifi-contracts@0.1.0 compile /home/runner/work/contracts/contracts
> hardhat compile && hardhat diamondABI

Solidity 0.8.13 is not fully supported yet. You can still use Hardhat, but some features, like stack traces, might not work correctly.

Learn more at https://hardhat.org/reference/solidity-support

Nothing to compile
No need to generate any newer typings.
ABI written to diamondABI/diamond.json


  AnyswapFacet
Adding facets...
Diamond cut tx:  0x2ea0482dc3cd16be37b629db4b3c79c960d8f983aac58bb17e06c0580125204d
Done.
Adding/Replacing facets...
Diamond cut tx:  0x306cd3de32a969bffdac56ef861466829301fa3327074e859834c6ebbf0a81c7
Done.
Adding/Replacing facets...
Diamond cut tx:  0xdca1905c85335372c114a5bf332871186cea3173d051f607d72478f5ac0755e0
Done.
Adding/Replacing facets...
Diamond cut tx:  0xd7a212259c2837608638de3a0e2491765c424f7f21e6f2882d8e351a6437ee89
Done.
     ✓ starts a bridge transaction using native token on the sending chain (1996ms)
     ✓ starts a bridge transaction using anyToken implementation on the sending chain (2994ms)
     ✓ starts a bridge transaction on the sending chain (3967ms)
     ✓ performs a swap then starts bridge transaction on the sending chain (3898ms)
     ✓ fails to perform a swap when the dex is not approved

  CBridgeFacet
Adding facets...
Diamond cut tx:  0x2944097eced500d96d794bdff64eafdbf126aa32a6268b5e57b1be7ddbe35186
Done.
Adding/Replacing facets...
Diamond cut tx:  0xee4d6d88746f3fb07ac9765d961d516bca8f189f40339d6e86fafd3b1d02758b
Adding facets...
Diamond cut tx:  0xde0220833264a5b1842f55f4a96f34a13d97b6e26d1b35a866a322a44d95b89f
Done.
Adding/Replacing facets...
Diamond cut tx:  0xb48237f5f622bd47a8c63a0a7104e8a0d8975edb22102f7f36ff86814dc79724
Done.
Adding/Replacing facets...
Diamond cut tx:  0xdc0bed44225c140dd010a0d5635a8b7daa993e0652bd806dcfd871459d46c13c
Done.
     ✓ starts a bridge transaction on the sending chain (17801ms)
     ✓ starts a bridge transaction on the sending chain with MATIC (13430ms)

  HyphenFacet
Adding facets...
Diamond cut tx:  0x4155832ed3ced51bbf88e203a3d88dcc8b186ea5caf90d279c98140fb039903c
Done.
Adding/Replacing facets...
Diamond cut tx:  0x89bb69e7de1b9a4c21617b938166058b582a959de653de20e2011983071b4487
Done.
Adding/Replacing facets...
Diamond cut tx:  0xcf0d1fe6aefea4f4966e51aab0e75603a00a06793fcc3c54b03f941a9c1b04f7
Done.
Adding/Replacing facets...
Diamond cut tx:  0xc6daca346dc2befaa9b8c0ab237431e307c83e1a0503ca3d0732fe5d89a0695c
Done.
     ✓ starts a bridge transaction on the sending chain (12143ms)
     ✓ performs a swap then starts bridge transaction on the sending chain (3692ms)

  NXTPFacet
Adding facets...
Diamond cut tx:  0x34c4b51837f1582228d4620c32d69db7f51eae58313145cad84bf34b35a246d1
Done.
Adding/Replacing facets...
Diamond cut tx:  0x3998709cab56ed754d8f208945541edb0da36437ceb2d84bea61b6fe3b937f85
Done.
Adding/Replacing facets...
Diamond cut tx:  0x5a442e1658b96cbd554bd7caf7af681d3a71f5939e7f90fd7d6e46328f7e5813
Done.
Adding/Replacing facets...
Diamond cut tx:  0x67ff70000a8d509501fec9f39af0581653528d7a1f1d50b16983cb2295e6aefb
Done.
     ✓ starts a bridge transaction on the sending chain (1564ms)
     ✓ performs a swap then starts bridge transaction on the sending chain (1663ms)
     ✓ fails to perform a swap when the dex is not authorized
     ✓ performs a swap with positive slippage then starts bridge transaction on the sending chain
     ✓ performs a swap with max approval and then does not approve for subsequent swaps (296ms)

  NXTPFacet (Paraswap)
Adding facets...
Diamond cut tx:  0xd9d51bf0056f04da2eaca0e94e18c6defa1fedb73a1395864f1ebcd31fed7200
Done.
Adding/Replacing facets...
Diamond cut tx:  0x0f7f1d35a5c21b83229f67f378310c76eaa3972c3c2005b95621f0a4f256c68c
Done.
Adding/Replacing facets...
Diamond cut tx:  0xc2ea73c94ae044fd164611d4d64bcf93f077910da1c643ff9ca9af7acc189fbc
Done.
Adding/Replacing facets...
Diamond cut tx:  0xb08bb7e53d2c02cdbb9bf57c2fb84e2828cc7f7898fc40190d3f7484dcfd387f
Done.
     ✓ performs a swap then starts bridge transaction on the sending chain (16824ms)


  28 passing (3m)

==forge tests

Running 5 tests for test/solidity/Facets/OwnershipFacet.t.sol:OwnershipFacetTest
[PASS] testFailNonOwnerCanTransferOwnership() (gas: 18829)
[PASS] testFailOnwershipTransferToNullAddr() (gas: 12572)
[PASS] testFailOwnerCanConfirmPendingOwnershipTransfer() (gas: 39508)
[PASS] testFailOwnershipTransferToSelf() (gas: 12767)
[PASS] testOwnerCanTransferOwnership() (gas: 38923)
Test result: ok. 5 passed; 0 failed; finished in 9.83s

Running 3 tests for test/solidity/Facets/Swapper.t.sol:SwapperTest
[PASS] testSingleSwap() (gas: 1736318)
[PASS] testSwapCleanup() (gas: 2641633)
[PASS] testSwapMultiInOne() (gas: 2608844)
Test result: ok. 3 passed; 0 failed; finished in 1.43s

Running 8 tests for test/solidity/Facets/DexManagerFacet.t.sol:DexManagerFacetTest
[PASS] testCanAddDEX() (gas: 84980)
[PASS] testCanApproveBatchFunctionSignature() (gas: 143517)
[PASS] testCanApproveFunctionSignature() (gas: 39742)
[PASS] testCanBatchAddDEXs() (gas: 182506)
[PASS] testCanBatchRemoveDEXs() (gas: 152603)
[PASS] testCanRemoveDEX() (gas: 69127)
[PASS] testFailAddZeroAddress() (gas: 12646)
```

```
[PASS] testFailBatchAddZeroAddress() (gas: 129879)
Test result: ok. 8 passed; 0 failed; finished in 3.65s

Running 2 tests for test/solidity/Facets/CBridgeFacet.t.sol:CBridgeFacetTest
[PASS] testCanBridgeTokens() (gas: 187910)
[PASS] testCanSwapAndBridgeTokens() (gas: 302933)
Test result: ok. 2 passed; 0 failed; finished in 20.97s
```

Code Coverage

The coverage report cannot be obtained at this moment due to tests that interact with the fork of different blockchains. It is highly recommended to solve this issue whenever it is possible and to have code coverage to at least 90%, in order to avoid functional bugs that are not necessarily security issues.

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
bc804440310ad19b9f32d77f8d4f3dc3011a10d361d5e2218ea72dec09f904a5 ./src/LiFiDiamond.sol
43d25ac3ed9c40b54f66bfa9d303f2f7274ed2359cc879a440c912765ed6ac3a ./src/Libraries/LibAsset.sol
c62733730a302097dc6abd6280bd6c3bf080bce8cc0232f9a300c6e285e4fb8e ./src/Libraries/LibDiamond.sol
264187cade829d22702d6982f18f6864b72394f4f6f0690119b86e78e3eb6d2f ./src/Libraries/LibStorage.sol
e1714bf58322707c5d120f404a5b0d15514383f73da72e1230b9ec4b454218f1 ./src/Libraries/LibSwap.sol
3b195f5b99259b7743cc16cc918359e3f01330a5b64ee17d373f8e70961f729b ./src/Libraries/LibUtil.sol
e49c4e4dc55539f5dace6e383c4ccacdf9c25914f0c38468cdc04ef2453fc5e7 ./src/Interfaces/IAnyswapRouter.sol
4cbbe146445aeaa78b2b4b9be152f6b136a471b188d1fb3cd8b0e18cc9c5ebd3 ./src/Interfaces/IAnyswapToken.sol
24a03b4e88288fb025101ec2122d1dc5fcc6cef848d147cb85914a1f3e565b15 ./src/Interfaces/ICBridge.sol
99c305bf84d7198c5335dc13069e9047b379c2dfe52d2f5840dcf8ace4954629 ./src/Interfaces/IDiamondCut.sol
d3842df6d2a93fde7bcd183a5e2ae71678eb67f31728e3e0e55a6ef81150c744 ./src/Interfaces/IDiamondLoupe.sol
0153181634f96c9c4e5b00a47c0219fca1f95f18b1748f45201c8b80a185da26 ./src/Interfaces/IERC165.sol
520b4640d29c8c405de9dadbc4d943dec4b7489b0117aa6da2349419698022c2 ./src/Interfaces/IERC173.sol
152e5c71974d7c02ade217ca442968b6894cf3533e357fd4e59aac2b736f0c56 ./src/Interfaces/IHopBridge.sol
240359b2dae1fcb10aa1697721ea237d5fd62fb1280e50d18ceb10bf3f18854a ./src/Interfaces/IHyphenRouter.sol
949a8775ad1ec221c7efbb07b11b9d69da69610bf8acf434a6ed8aebf4a003d7 ./src/Interfaces/ILiFi.sol
ebda310fbda4b8e877287cfaad87865d52f8c60908433069c41d75954211eccd ./src/Interfaces/ITransactionManager.sol
1d84ffedf8a818d2ffb3ea03369a43e609aabad35fad01d7141cd56be5f545bb ./src/Helpers/ReentrancyGuard.sol
68e9613994b8487b89b79a859bd096eb46916de6b891e4b33366175d52eec26d ./src/Helpers/Swapper.sol
055458da6545d101351875a44606a355c9ca23c6adcdb532443d27e97997cb4c ./src/Facets/AnyswapFacet.sol
b1ea7a6eb6590af11e295096ed2a51e23d0bc4b2cf275b834c49e8a5b42abb13 ./src/Facets/CBridgeFacet.sol
66f96b208716556233d95e77ea638685e7482339c2417b73a912e766417bb478 ./src/Facets/DexManagerFacet.sol
0c6b6c745374176b905e005cfc8a36fde45c8ca46fc0349124e8a736cf538f75 ./src/Facets/DiamondCutFacet.sol
24743b30ec81be13e9375fd9746db9b95067d81801eeb5124e07d898a8fa68fa ./src/Facets/DiamondLoupeFacet.sol
05473d9635775967a7d77e9b65a2fd80656f6ad05a7c632f8c62fa521031bda7 ./src/Facets/GenericSwapFacet.sol
d4a18d8cd0fc24b536ec38f8c41a1445bc909f2216a718aeb13de0009bbf380b ./src/Facets/HopFacet.sol
45d71e3d0de1dce2213cdf1e80d273569e89b72600ac343315abc51d11f4bb1e ./src/Facets/HyphenFacet.sol
63370850f47b9fb86878c51040ec9884317a34de2e39ef59e8f5cd3217f025e8 ./src/Facets/NXTPFacet.sol
8f4f3bb7c165082577f11c1df5f60ee661ee6cbd0b31aba0af417d523f87b048 ./src/Facets/OwnershipFacet.sol
e15da9f6ea08ac5f7b0188e1ed8cb620636f049643b2897e99f93f11e7388718 ./src/Facets/WithdrawFacet.sol
420e720cee73c9396d57ae0446797ded614281d9e666daa457237fb83a86da8c ./src/Errors/GenericErrors.sol
```

Tests

```
4f66b6debfcc070ea2b44420006642b5cb92e98a91ef533e4d3dc009c9d2b530 ./test/chai-setup.ts
fa849c28501ac0e2bf913ff40254bab3c5c2a09ca2aade48004b426d5a2a1395 ./test/utils/index.ts
2d11ce95a2962bd3a7e98d9dbde9f001fd7cb3dc119ec52a962f2cd98ac8ce25 ./test/solidity/utils/Console.sol
ebbe6a5770af072315cb1b93115cd90737a8593b338b89ba16cad1aa53f9b697 ./test/solidity/utils/DiamondTest.sol
8f1573158e443a388137da6736802bcb7149916b5ec709e91c492d42a2e070b0 ./test/solidity/utils/Interfaces.sol
587005023a64335a40772dd35d71420d6d709155e5d405f163fb2f8ee5dcc8b3 ./test/solidity/utils/TestAMM.sol
7e310707e52b8c70f49f88482375b2542fe7df58ef70c9a53434c12a87822218 ./test/solidity/utils/TestToken.sol
7720c22a3e3b36de7073bdb8e0b8667dd0aa4ff1ecc28f497b3ca105d3d71ee0 ./test/solidity/utils/Utilities.sol
03f5922e2a8cfa88970d01ab344eaf0a46189dfb4162f0f80ff7b50010a88f7a ./test/solidity/Facets/CBridgeFacet.t.sol
f5be4d0614b54ee4c6a672f3fe3294a4575aae17c99629e15e339d6c6eb405b1 ./test/solidity/Facets/DexManagerFacet.t.sol
1188686c97517a5fc270ae1cbe585adf6d9471012dc934b6a2bd93c4ce8597e0 ./test/solidity/Facets/OwnershipFacet.t.sol
cfbf2e4da41fa0376fa2f8f472645a6a7c51f2941c7be24e6cc7d0fd48d2ea12 ./test/solidity/Facets/Swapper.t.sol
d84f74fea7ec2471c6bd65a2680b83744b8f2d0661c0849aea89edd77af73a68 ./test/fixtures/nxtp.ts
3191fcadecd964a5d5ee9316fee4ffadbda9b8a129af09e021124c24e41dbac0 ./test/facets/AnyswapFacet.test.ts
df9927877bce2c38b69fbfcd66eaaeb67ad42062b51cce69d350f3f40027d0b7 ./test/facets/CBridgeFacet.test.ts
ae02a44ee76cb141895a92f46de7603a5d559a7e6ff7a5ea0dcc12a79b607e63 ./test/facets/GenericSwapFacet.test.ts
84132cd24c59e748b8f20ad7152736c23b137cf8eea5420cfdfe39b11c4cfa5e ./test/facets/HopFacetL1.test.ts
035f0f58e91e012b376616607d086d7993968091f397f852d3fd2038201553d9 ./test/facets/HopFacetL2.test.ts
31404cd68fa8534846fa796e6fe768ea27c3e10642e9ed100428f55dfb2ab2ec ./test/facets/HyphenFacet.test.ts
2e7753d4ab0957eb773e1528374375d15c1ea19eb049f47a689141089ed2a8ae ./test/facets/NXTPFacet.test.ts
b5b1fca38f148e4a9a5b020840f248492d76f017ccdad61151b29323076b48b4 ./test/facets/NXTPFacetParaswap.test.ts
```

Changelog

- 2022-04-22 - Initial report
- 2022-05-05 - Final report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.