CERTIK

Security Assessment

Gridex

CertiK Verified on Feb 11th, 2023

CertiK Verified on Feb 11th, 2023

Gridex

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Ethereum	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 02/11/2023	N/A

CODEBASE

https://github.com/GridexProtocol/core

...View All

COMMITS

base: 41ee73d3569eb9905d3fc2cc331ee8963ca35144

update1: c0b6db818a0aec037245db4d14f27356e0aff9d6

update2: 48658e46697ecc34d2cad14478fca9971ab21414

...View All

Vulnerability Summary

9	9	0	0	0	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
1	Major	1 Resolved	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
1	Minor	1 Resolved	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
7	Informational	7 Resolved	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | GRIDEX

CODEBASE | GRIDEX

Repository

https://github.com/GridexProtocol/core

Commit

base: 41ee73d3569eb9905d3fc2cc331ee8963ca35144
update1: c0b6db818a0aec037245db4d14f27356e0aff9d6
update2: 48658e46697ecc34d2cad14478fca9971ab21414
update3: e0974f17be0e0991b7f7780184301bbc7cead533
update4: e08271085ab3e821f4ccd5c91e49376ba637ee1d

AUDIT SCOPE | GRIDEX

30 files audited ● 1 file with Acknowledged findings ● 17 files with Resolved findings ● 12 files without findings

ID	Repo	Commit	File	SHA256 Checksum
BMP	GridexProtocol/core	41ee73d	contracts/libraries/BoundaryMath.sol	85f15802c6be0fd50f8632d8433cccc9db6f4b39f9e566d1fa78de54b84bdd35
IGD	GridexProtocol/core	41ee73d	contracts/interfaces/IGridDeployer.sol	fd67ee914642ee07a172409d38e7fa690d73e5e519a343d90038c57da8363e96
IGP	GridexProtocol/core	41ee73d	contracts/interfaces/IGridParameters.sol	b8244da33db171e5533d77bef4a35703df1de2cebea5f35cb38ce6a26c778cf1
IPO	GridexProtocol/core	41ee73d	contracts/interfaces/IPriceOracle.sol	d03d580bd762ca1330f7ca7912b63293247c4456a67aaab25e9ab16670f55de8
ITC	GridexProtocol/core	41ee73d	contracts/interfaces/ITradingConfig.sol	3d408b8f2cc56f9699a402b5151de90671de089c3007afc9e4fc867c04152e7c
BBG	GridexProtocol/core	41ee73d	contracts/libraries/BoundaryBitmap.sol	118695ab983b5d8567e2ab8a98a99d4c57118e5840f6dfb2434890d2a347b72c
BGP	GridexProtocol/core	41ee73d	contracts/libraries/BundleMath.sol	9d751621c3501102e4b50005ca3314ec6e04e6ff8bbb30852d1c7edfff3f8cef
CVG	GridexProtocol/core	41ee73d	contracts/libraries/CallbackValidator.sol	5c86aa1dd3889db5fcd17a80214b226fc784f268ab9db82df97c1d2459467831
FMG	GridexProtocol/core	41ee73d	contracts/libraries/FeeMath.sol	8448b3af42497f5f74e53424ee3e6c551f51356945108d22a893d608a7990542
GAG	GridexProtocol/core	41ee73d	contracts/libraries/GridAddress.sol	309f6072dc843d7aa3edfc3a02f3b5498db07f8d8af9edc1660d39b0abe5eff8
SMG	GridexProtocol/core	41ee73d	contracts/libraries/SwapMath.sol	5b7d38985366704d6e2f1be8697c6ad985919fd9460a3b9b45c1da8384d9143f
UMG	GridexProtocol/core	41ee73d	contracts/libraries/Uint128Math.sol	3ed5947a7c629898d0f692d5bc0ac9c9689da9e0b58dde82b4693446e89f08ee
UMP	GridexProtocol/core	41ee73d	contracts/libraries/Uint160Math.sol	cc089692343d1cc36eaf196046d7a528d153abd55ba20e82f1d57c22fcd92675

ID	Repo	Commit	File	SHA256 Checksum
GGP	GridexProtocol/core	41ee73d	contracts/Grid.sol	245bb370a16615d4a6e5db22ab419781c62b3de2123c753f558788977f6c738d
GDG	GridexProtocol/core	41ee73d	contracts/GridDeployer.sol	c1a88bc370ccde066fa511c8c2e1124b6f861539fbe3365544be66964e82d2f7
GFG	GridexProtocol/core	41ee73d	contracts/GridFactory.sol	4befbded94c2f4ef29433ce90f1950b4ee66dc09cd475789c8386f014afb79f9
POG	GridexProtocol/core	41ee73d	contracts/PriceOracle.sol	6adf00d2b2b7d65850569cf5aa5872b67fa93936527aed841f95a4b64ad40e3b
TCG	GridexProtocol/core	41ee73d	contracts/TradingConfig.sol	2932cecfdacf0cb509001b1ce1afaa05f3f235f84a0d52b974584ea25bcc0adb
IGG	GridexProtocol/core	41ee73d	contracts/interfaces/IGrid.sol	66686af896d9de715e6dc700c0d447459d81bdb574925c1d29fdeb150de0f616
IGE	GridexProtocol/core	41ee73d	contracts/interfaces/IGridEvents.sol	3cce9d7148ed5cbc67e458b8f8d5e2634ed71b9d47d659e608399796cc40283b
IGF	GridexProtocol/core	41ee73d	contracts/interfaces/IGridFactory.sol	3564112321f5024e28cb372581e7a2dfcc8de580a336a5635507d0e391d05170
IGS	GridexProtocol/core	41ee73d	contracts/interfaces/IGridStructs.sol	6b14bdd157924b36dec1901f8dae934ef188237689735470cf0585b3aa381ce8
IWE	GridexProtocol/core	41ee73d	contracts/interfaces/IWETHMinimum.sol	e91ecc404ace2c85c3a4d298b7f8d9b12669b80b0eb9ef5037937524ae5da096
IGC	GridexProtocol/core	41ee73d	contracts/interfaces/callback/IGridFlashCallback.sol	9ebe273980355dd9541a1189ad1899de77d87d7824517310b53bd18c11160284
IGM	GridexProtocol/core	41ee73d	contracts/interfaces/callback/IGridPlaceMakerOrderCallback.sol	ceb1c760c0ac05168504044fbecc2d8b1d3a682a5fc1137a56fe251e158a7dd8
ISC	GridexProtocol/core	41ee73d	contracts/interfaces/callback/IGridSwapCallback.sol	e7b174ea22a86e5c4bbc96d10a73d9c59eb9250e44fbb16a6f166d5e25104d0a
BMG	GridexProtocol/core	41ee73d	contracts/libraries/BitMath.sol	0323119f4e2fc91b856925b02da2155efce81bacadff8e93417e34091dd763bd

ID	Repo	Commit	File	SHA256 Checksum
FPX	GridexProtocol/core	41ee73d	contracts/libraries/FixedPointX128.sol	8a78fa35dbb7d66818c3774a76a161cd2c10fa8db399d17cc426267f6061e825
FPG	GridexProtocol/core	41ee73d	contracts/libraries/FixedPointX192.sol	316370e4d54377a96557c754dee8a1621eb270d3f237b21383e76400334ad4cf
FPP	GridexProtocol/core	41ee73d	contracts/libraries/FixedPointX96.sol	244e9f70ac61e353c5d99828292ba29fa306453937b5d17228d91958412e8926

APPROACH & METHODS | GRIDEX

This report has been prepared for Gridex to discover issues and vulnerabilities in the source code of the Gridex project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

FINDINGS | GRIDEX

9	0	1	0	1	7
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Gridex. Through this audit, we have uncovered 9 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
TCG-01	**Centralization Risks In TradingConfig.Sol**	**Centralization / Privilege**	**Major**	**Resolved**
BMG-01	Possible Overflow	Mathematical Operations	Minor	Resolved
BBG-01	Comments For `nextInitializedBoundary()` And `nextInitializedBoundaryWithinOneWord()`	Inconsistency	Informational	Resolved
GGP-01	Missing Override Specifier	Inconsistency	Informational	Resolved
GPB-01	Typos	Coding Style	Informational	Resolved
GPB-02	Unlocked Compiler Version	Language Specific, Compiler Error	Informational	Resolved
GPB-03	Missing Parameter In Natspec Comments	Inconsistency, Coding Style	Informational	Resolved
GRI-01	`channel` May Not Be Aware They Must Collect `channelFees` Before Overflow	Coding Style	Informational	Resolved
IGE-01	`Swap` Event Should Emit `channel` Fee	Coding Style	Informational	Resolved

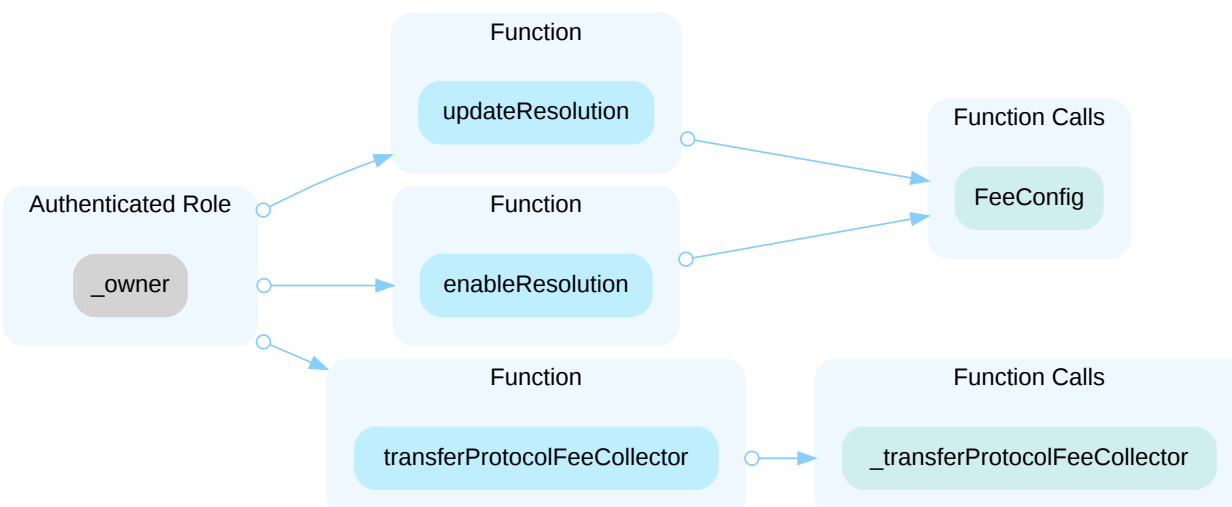
TCG-01 | CENTRALIZATION RISKS IN TRADINGCONFIG.SOL

Category	Severity	Location	Status
Centralization / Privilege	**Major**	**contracts/TradingConfig.sol (base): 29, 45, 59**	**Resolved**

Description

In the contract `TradingConfig`, the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and do the following:

- Enable any positive and non-zero resolution that has not been enabled yet, with any `takerFee` and `makerFee` that satisfies the following conditions:
  - `takerFee` must be greater than 0 and less than or equal to `1e4`.
  - `makerFee` must be negative or 0 and be such that `-makerFee <= takerFee`.
- Update any enabled resolutions fees, with any `takerFee` and `makerFee` that satisfies the conditions above. Taker and maker fees are immutable in the grid contract, so these updated fees will only apply to new grids created after `updateResolution()` is called.
- Change the `protocolFeeCollector` to a wallet they control in order to collect any protocol fees.



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts

with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

`[CertiK]` : The client removed `TradingConfig.sol` .

BMG-01 | POSSIBLE OVERFLOW

Category	Severity	Location	Status
Mathematical Operations	Minor	contracts/libraries/BundleMath.sol (update3): 34	Resolved

Description

In `updateForTaker`, the following equation is used inside an unchecked block:

```
parameters.amountInUsed = uint128((uint256(parameters.amountOutUsed) * amountIn) /
amountOut);
```

Where `parameters.amountOutUsed` is a `uint128` and `amountIn` is a `uint256`. As these two are multiplied together in an unchecked block, they may overflow.

Recommendation

We recommend either checking for overflow in this case, or ensuring that the `amountIn` is small enough it will never cause an overflow.

Alleviation

`[CertiK]`: The client made the recommended changes in commit: e08271085ab3e821f4ccd5c91e49376ba637ee1d.

BBG-01 COMMENTS FOR `nextInitializedBoundary()` AND `nextInitializedBoundaryWithinOneWord()`

Category	Severity	Location	Status
Inconsistency	Informational	contracts/libraries/BoundaryBitmap.sol (base): 35~36, 85~86	Resolved

Description

The comments for `nextInitializedBoundary()` are missing the parameter `boundaryLower`. In addition, for both functions `nextInitializedBoundary()` and `nextInitializedBoundaryWithinOneWord()`, the parameter `lte` can cause confusion as it searches strictly to the left, that is strictly less than the starting boundary.

Recommendation

We recommend adding a comment for the `boundaryLower` parameter and changing the comment and name of `lte` to reflect that it searches strictly to the left.

Alleviation

`[CertiK]`: The client added a comment for the `boundaryLower` in commit: 9bb26c2285124f10384f07ff9de82630b237142. However, the naming of `lte` was kept with the client stating the following:

`[Gridex]`: "The naming of lte should be kept, there are already clear comments."

GGP-01 | MISSING OVERRIDE SPECIFIER

Category	Severity	Location	Status
Inconsistency	Informational	contracts/Grid.sol (base): 82	Resolved

Description

The function `syncFee()` does not have the override specifier. It should be noted that since version 0.8.8, a function that overrides only a single interface function does not require the override specifier (see doc). However, all other instances of this in the codebase contain the override specifier.

Recommendation

We recommend adding the override specifier to `syncFee()` or removing the override specifier from all other functions this applies to for consistancy.

Alleviation

`[CertiK]` : The client made the recommended changes in commit: a355023a388406000d438e8554cf51b7ff7fa529.

GPB-01 | TYPOS

Category	Severity	Location	Status
Coding Style	Informational	contracts/Grid.sol (base): 218, 225, 246, 473; contracts/GridDeployer.sol (base): 18; contracts/libraries/SwapMath.sol (base): 125	Resolved

Description

In the file `GridDeployer`, there are grammatical errors that can cause confusion:

- On line 18, "TThe" is spelled incorrectly and should be spelled "The".
- On line 18, `token1` is the second token (not the first) in the grid, after sorting by address.

In the file, `Grid`, there are some grammatical errors that can cause confusion:

- On line 218, "tokens to be recieve" would be clearer as "tokens to be recieved".
- On line 225, "token to pay failed" would be clearer as "token payment failed" or "token pay failed".
- On line 246, "we locks the grid before swap" would be clearer as "we lock the grid before swap".

In the file, `SwapMath.sol`,

- On line 125, the commented equation is not equivalent. The equation should not be multiplied by `FixedPointX96.Q`.

Recommendation

We recommend fixing these typos or unclear comments to enable all reviewers to prevent any confusion.

Alleviation

`[CertiK]`: The client made the recommended changes in the commit: 60b466420857a3c0576d4af1b8fc877cecd20f08.

GPB-02 | UNLOCKED COMPILER VERSION

Category	Severity	Location	Status
Language Specific, Compiler Error	Informational	contracts/Grid.sol (base): 2, 66~68; contracts/GridDeployer.sol (base): 2; contracts/GridFactory.sol (base): 2; contracts/PriceOracle.sol (base): 2; contracts/TradingConfig.sol (base): 2	Resolved

Description

The contracts cited have an unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging, as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Until version `0.8.8`, it was not possible to read immutable variables during contract creation time. However, starting at this version, immutable variables can be read at construction time once they are initialized (see doc). As the variables `resolution` and `tradingConfig` are immutable variables that are initialized and then read in the `constructor()`, a compiler version `0.8.8` or greater must be used.

Recommendation

We recommend changing the `pragma` to a locked compiler version `0.8.8` or greater.

Alleviation

`[CertiK]`: The client fixed the compiler issues by locking it to compile at 0.8.9 in commit: dd3185d7561c300563b69cfab11fe739add357f4.

GPB-03 | MISSING PARAMETER IN NATSPEC COMMENTS

Category	Severity	Location	Status
Inconsistency, Coding Style	Informational	contracts/interfaces/IGridDeployer.sol (base): 33~35; contracts/interfaces/IPriceOracle.sol (base): 44; contracts/interfaces/ITradingConfig.sol (base): 19~21; contracts/libraries/BoundaryMath.sol (base): 13, 17, 21, 25, 212, 221; contracts/libraries/BundleMath.sol (base): 18, 54, 61, 76; contracts/libraries/CallbackValidator.sol (base): 7; contracts/libraries/GridAddress.sol (base): 15, 21; contracts/libraries/SwapMath.sol (base): 21, 27, 55, 95, 168, 206, 217, 253, 269, 293, 311; contracts/libraries/Uint128Math.sol (base): 5, 9; contracts/libraries/Uint160Math.sol (base): 5, 9	Resolved

Description

Inside `IGridDeployer` the Natspec comments of the function `parameters` does not include the description of the `address` parameters.

Inside `IPriceOracle` the Natspec comments of the function `gridPriceData` does not include the description of the `index` parameter.

Inside `ITradingConfig` the Natspec comments of the event `ProtocolFeeCollectorTransferred` does not include the description of the `index` parameter.

Inside `BoundaryMath` the Natspec comments for the functions:

- `isValidBoundary` does not include the description for input parameters.
- `isInRange` does not include the description for input parameters.
- `getPriceX96AtBoundary` does not include the description for input parameters.
- `getBoundaryLowerAtBoundary` does not include the description for input parameters.
- `rewriteToValidBoundaryLower` does not include the description for input parameters.

Inside `BundleMath` the Natspec comments for the functions:

- `updateForTaker` does not include the description for the return values.
- `addLiquidity` does not include the description for input parameters.
- `addLiquidityWithAmount` does not include the description for input parameters.

- `removeLiquidity` does not include the description for return values.

Inside `CallbackValidator` the Natspec comments for the function `validate` does not include the description of `gridFactory` and `gridKey`.

Inside `GridAddress` the Natspec comments for the functions:

- `gridKey` does not include the description for input parameters or return values.
- `computeAddress` does not include the description for input parameters or return values.

Inside `SwapMath` the Natspec comments for the functions:

- `computeSwapStep` does not include the description for input parameters or return values.
- `computeSwapStepForExactIn` does not include the description for input parameters or return values.
- `_computeSwapStepForExactIn` does not include the description for input parameters or return values.
- `computeSwapStepForExactOut` does not include the description for input parameters or return values.
- `_priceInRange` does not include the description for input parameters or return values.
- `_computePriceNextX96` does not include the description for input parameters or return values.
- `_computeAmountInAndFeeAmount` does not include the description for input parameters or return values.
- `_computeAmountOutForPriceLimit` does not include the description for input parameters or return values.
- `_divUpForPriceX96` does not include the description for input parameters or return values.

Inside `Uint128Math` the Natspec comments for the functions `minUint128` and `maxUint128` do not include the description of `a` and `b`.

Inside `Uint160Math` the Natspec comments for the functions `minUint160` and `maxUint160` do not include the description of `a` and `b`.

Recommendation

We recommend adding the description of input parameters and return values to increase readability for users.

Alleviation

`[CertiK]`: The client made the recommended changes in the commit: f89bffb9167d3f3672d8372f76a02d919620efc5.

GRI-01 `channel` MAY NOT BE AWARE THEY MUST COLLECT `channelFees` BEFORE OVERFLOW

Category	Severity	Location	Status
Coding Style	Informational	contracts/Grid.sol (update2): 296~298	Resolved

Description

Any user can now specify a `channel` to receive 80 percent of the protocol fees and overflow is permitted for them. However, a `channel` can be any address and may not be aware they must collect their `channelFees` prior to overflow.

Recommendation

We recommend either checking for overflow in the logic for `channelFees` or providing clear documentation to your community that `channelFees` must be claimed prior to overflow. In addition, we recommend changing the comment to reflect that it updates the `channelFees` as well.

Alleviation

`[CertiK]` : The client removed the functionality for protocol fees.

IGE-01 | `Swap` EVENT SHOULD EMIT `channel` FEE

Category	Severity	Location	Status
Coding Style	Informational	contracts/interfaces/IGridEvents.sol (update2): 69~76	Resolved

Description

A `swap()` now sends 80 percent of the protocol fees to the input `channel`, which can be set to any address by the caller.

Recommendation

We recommend also emitting the `channel` in the `Swap` event.

Alleviation

`[CertiK]`: The client removed the functionality for protocol fees.

OPTIMIZATIONS | GRIDEX

ID	Title	Category	Severity	Status
BMP-01	Logarithm Refinement Optimization	Gas Optimization	Optimization	Acknowledged
GFG-01	Checks Can Be Performed Earlier	Gas Optimization	Optimization	Resolved
GPU-01	Unnecessary Use Of SafeMath	Gas Optimization	Optimization	Resolved
IGP-01	Struct Optimization	Gas Optimization	Optimization	Resolved
TCG-02	Unused State Variable	Gas Optimization	Optimization	Resolved

BMP-01 | LOGARITHM REFINEMENT OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/libraries/BoundaryMath.sol (base): 192~196, 201~202	Acknowledged

Description

The function `getBoundaryAtPrice()` uses 14 refinements for `log_2`. However, 13 refinements can be used instead provided the error bounds are re-calculated. This doc goes through the derivation of these values and can be used to calculate the new error bounds. Using 13 instead of 14 refinements saves around 41 gas.

Recommendation

We recommend re-calculating these values to use 13 refinements in order to reduce gas costs.

Alleviation

`[CertiK]`: The client acknowledged the finding but did not make any changes.

GFG-01 | CHECKS CAN BE PERFORMED EARLIER

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/GridFactory.sol (base): 35, 36~37, 38~39, 42	Resolved

Description

The variables `token0` and `token1` are assigned before checks on `tokenA`, `tokenB`, and `takerFee` are made. If one of these checks fails, then additional gas is paid to assign these variables unnecessarily.

Recommendation

We recommend assigning the variables `token0` and `token1` after these checks are performed to reduce the gas cost when one of these checks fails.

Alleviation

`[CertiK]`: The client made the recommended changes in commit: 8db7d1075719cd1444726b065eb78121993aabe3.

GPU-01 | UNNECESSARY USE OF SAFEMATH

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/libraries/FeeMath.sol (base): 9, 10; contracts/libraries/SwapMath.sol (base): 12~13	Resolved

Description

The `SafeMath` library is used unnecessarily. With Solidity compiler versions 0.8.0 or newer, arithmetic operations will automatically revert in case of integer overflow or underflow.

Recommendation

We recommend removing the usage of the `SafeMath` library and using the built-in arithmetic operations provided by the Solidity programming language.

Alleviation

`[CertiK]` : The client made the recommended changes in commit: d5370c2c6dc3d71717ab72fd08f9bf9655ed00bc.

IGP-01 | STRUCT OPTIMIZATION

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/interfaces/IGridParameters.sol (base): 33	Resolved

Description

The `SwapState` struct is not tightly packed and could save gas. `SwapState` uses an entire 32 byte slot to store a boolean. If the boolean is packed with a `uint160`, this saves 1 storage slot from being used.

Recommendation

We recommend rearranging the bool value inside the struct to conserve gas.

Alleviation

`[CertiK]`: The client acknowledged the finding and explained it is to save gas.

TCG-02 | UNUSED STATE VARIABLE

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/TradingConfig.sol (base): 9	Resolved

Description

The constant variable `MAX_FEE` is defined in the contract `TradingConfig.sol`, but it is is never used in the contract.

Recommendation

We recommend removing or implementing the unused variable.

Alleviation

`[CertiK]`: The client made the recommended changes in commit: 1bef11ed050bec3182e9b04e783090dfc5042990.

APPENDIX | GRIDEX

Finding Categories

Categories	Description
Centralization / Privilege	Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.
Gas Optimization	Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.
Mathematical Operations	Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.
Language Specific	Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.
Coding Style	Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.
Inconsistency	Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.
Compiler Error	Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.