



Security Assessment Report

CompoundX

12 May 2026

This security assessment report was prepared by SolidityScan.com, a cloud-based Smart Contract Scanner.



Verified Report

This automated audit report has been verified by the SolidityScan team. To learn more about our published reports [click here](#).

Table of Contents

01 Vulnerability Classification and Severity

02 Executive Summary

03 Threat Summary

04 Findings Summary

05 Vulnerability Details

ADMIN FEE CHARGED WITHOUT REDUCING CREDITED PRINCIPAL CAUSES SYSTEMIC UNDER-COLLATERALIZATION

REWARDS CAN BE RETROACTIVELY BOOSTED BY CURRENT ROI TIER WHEN AUTO-CALCULATION IS SKIPPED

ADDING A RETURN STATEMENT WHEN THE FUNCTION DEFINES A NAMED RETURN VARIABLE IS REDUNDANT

HARD-CODED ADDRESS DETECTED

BLOCK VALUES AS A PROXY FOR TIME

CONSIDER USING UINT48 FOR TIME-RELATED VARIABLES

CONSTRUCTORS SHOULD EMIT AN EVENT

IF-STATEMENT REFACTORING

MISSING @AUTHOR IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS

MISSING @AUTHOR IN NATSPEC COMMENTS FOR CONTRACT DECLARATION

MISSING @DEV IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS

MISSING @DEV IN NATSPEC COMMENTS FOR CONTRACT DECLARATION

MISSING @DEV IN NATSPEC COMMENTS FOR FUNCTIONS

MISSING @INHERITDOC ON OVERRIDE FUNCTIONS

MISSING NATSPEC COMMENTS IN SCOPE BLOCKS

MISSING NATSPEC DESCRIPTIONS FOR PUBLIC VARIABLE DECLARATIONS

MISSING @NOTICE IN NATSPEC COMMENTS FOR CONSTRUCTORS

MISSING @NOTICE IN NATSPEC COMMENTS FOR FUNCTIONS

MISSING @NOTICE IN NATSPEC COMMENTS FOR MODIFIERS

MISSING UNDERSCORE IN NAMING VARIABLES

NAME MAPPING PARAMETERS

VARIABLES SHOULD BE IMMUTABLE

ARRAY LENGTH CACHING

ASSIGNING TO STRUCTS CAN BE MORE EFFICIENT

AVOID ZERO-TO-ONE STORAGE WRITES

CHEAPER CONDITIONAL OPERATORS

CHEAPER INEQUALITIES IN IF()

CHEAPER INEQUALITIES IN REQUIRE()

DEFAULT INT VALUES ARE MANUALLY RESET

DEFINE CONSTRUCTOR AS PAYABLE

EMIT USED IN LOOP

FUNCTIONS CAN BE IN-LINED

FUNCTION SHOULD RETURN STRUCT

GAS INEFFICIENCY DUE TO MULTIPLE OPERANDS IN SINGLE IF/ELSEIF CONDITION

GAS OPTIMIZATION FOR STATE VARIABLES

GAS OPTIMIZATION IN DECREMENTS

GAS OPTIMIZATION IN INCREMENTS

LONG NUMBER LITERALS

LONG REQUIRE/REVERT STRINGS

NAMED RETURN OF LOCAL VARIABLE SAVES GAS AS COMPARED TO RETURN STATEMENT

PUBLIC CONSTANTS CAN BE PRIVATE

SMALLER DATA TYPES COST MORE

SPLITTING REQUIRE STATEMENTS

STORAGE VARIABLE CACHING IN MEMORY

SUPERFLUOUS EVENT FIELDS

UNNECESSARY CHECKED ARITHMETIC IN LOOP






UNNECESSARY DEFAULT VALUE INITIALIZATION

05 Scan History

06 Disclaimer

01. **Vulnerability** Classification and Severity

Description

To enhance navigability, the document is organized in descending order of severity for easy reference. Issues are categorized as  *Fixed*,  *Pending Fix*, or  *Won't Fix*, indicating their current status.  *Won't Fix* denotes that the team is aware of the issue but has chosen not to resolve it. Issues labeled as  *Pending Fix* state that the bug is yet to be resolved. Additionally, each issue's severity is assessed based on the risk of exploitation or the potential for other unexpected or unsafe behavior.

- **Critical**

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

- **Medium**

The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.

- **Informational**

The issue does not affect the contract's operational capability but is considered good practice to address.

- **High**

High-severity vulnerabilities pose a significant risk to both the Smart Contract and the organization. They can lead to user fund losses, may have conditional requirements, and are challenging to exploit.

- **Low**

The issue has minimal impact on the contract's ability to operate.

- **Gas**

This category deals with optimizing code and refactoring to conserve gas.

02. Executive Summary



CompoundX

Published on 12 May 2026

0x8da44609d55ae3372c672d66be0e12776c609731

<https://bscscan.com/address/0x8da44609d55ae3372c672d66...> 

Language

Solidity

Audit Methodology

Static Scanning

Contract Type

-

Website

-

Publishers/Owner Name

-

Organization

-

Contact Email

-



Security Score is GREAT

The SolidityScan score is calculated based on lines of code and weights assigned to each issue depending on the severity and confidence. To improve your score, view the detailed result and leverage the remediation solutions provided.

This report has been prepared for CompoundX using SolidityScan to scan and discover vulnerabilities and safe coding practices in their smart contract including the libraries used by the contract that are not officially recognized. The SolidityScan tool runs a comprehensive static analysis on the Solidity code and finds vulnerabilities ranging from minor gas optimizations to major vulnerabilities leading to the loss of funds. The coverage scope pays attention to all the informational and critical vulnerabilities with over 700+ modules. The scanning and auditing process covers the following areas:

Various common and uncommon attack vectors will be investigated to ensure that the smart contracts are secure from malicious actors. The scanner modules find and flag issues related to Gas optimizations that help in reducing the overall Gas cost. It scans and evaluates the codebase against industry best practices and standards to ensure compliance. It makes sure that the officially recognized libraries used in the code are secure and up to date.

The SolidityScan Team recommends running regular audit scans to identify any vulnerabilities that are introduced after CompoundX introduces new features or refactors the code.

03. Threat Summary

Threat Score ⓘ



THREAT SUMMARY

Your smart contract has been assessed and assigned a **Low Risk** threat score. The score indicates the likelihood of risk associated with the contract code.

✓ Contract's source code is verified.

Source code verification provides transparency for users interacting with smart contracts. Block explorers validate the compiled code with the one on the blockchain. This also gives users a chance to audit the contracts, ensuring that the deployed code matches the intended functionality and minimizing the risk of malicious or erroneous contracts.

✓ The contract cannot mint new tokens.

Minting functions are often utilized to generate new tokens, which can be allocated to specific addresses, such as user wallets or the contract owner's wallet. This feature is commonly employed in various decentralized finance (DeFi) and non-fungible token (NFT) projects to facilitate token issuance and distribution. The Presence of Minting Function module is designed to quickly identify the presence and implementation of minting functions in a smart contract. Mint functions play a crucial role in creating new tokens and transferring them to the designated user's or owner's wallet. This process significantly contributes to increasing the overall circulation of the tokens within the ecosystem.

✓ The tokens cannot be burned in this contract.

The token contract incorporates a burn function that enables the intentional reduction of token amounts, consequently diminishing the total supply. The execution of this burn function contributes to the creation of scarcity within the token ecosystem, as the overall availability of the token decreases.

The contract can be compiled with a more recent Solidity version

The contract should be written using the latest Solidity pragma version as it comes with numerous bug fixes. Utilizing an outdated version exposes the contract to vulnerabilities associated with known issues that have been addressed in subsequent updates. Therefore, it is essential to stay current with the latest Solidity version to ensure the robustness and security of the contract against potential vulnerabilities.

This is not a proxy-based upgradable contract.

The Proxy-Based Upgradable Contract module is dedicated to identifying the presence of upgradeable contracts or proxy patterns within a smart contract. The utilization of upgradeable contracts or proxy patterns enables contract owners to make dynamic changes to various aspects, including functions, token circulation, and distribution, without requiring a complete redeployment of the contract.

Owners cannot blacklist tokens or users.[Ownership Renounced]


This module is designed to identify whether the owner of a smart contract has the capability to blacklist specific tokens or users. In a scenario where owners possess the authority to blacklist, all transactions related to the blacklisted entities will be immediately halted. Ownership privileges that include the ability to blacklist tokens or users can be a critical feature in certain use cases, providing the owner with control over potential malicious activities, compliance issues, or other concerns. However, in situations where this authority is abused or misapplied, it can lead to unintended consequences and user dissatisfaction.

Is ERC-20 token.

A token is expected to adhere to the established standards of the ERC-20 token specification, encompassing the inclusion of all necessary functions with standardized names and arguments as defined by the ERC-20 standard.

This is not a Pausable contract.

Pausable contracts refer to contracts that can be intentionally halted by their owners, temporarily preventing token holders from engaging in buying or selling activities. This pause mechanism allows contract owners to exert control over the token's functionality, introducing a temporary suspension in trading activities for various reasons such as security concerns, updates, or regulatory compliance adjustments.



Critical functions that add, update, or delete owner/admin addresses are not detected. [Ownership Renounced]

A smart contract within the Web3 ecosystem that incorporates critical administrative functions can potentially compromise the transparency and intended objectives of the contract. It is imperative to conduct a thorough examination of these functions, especially in the realm of Web3 smart contracts. Minimizing administrative functions in a token contract within the Web3 framework can significantly reduce the likelihood of complications and enhance overall efficiency and clarity.



The contract cannot be self-destructed by owners.

The SELFDESTRUCT opcode is a critical operation in Ethereum smart contracts, allowing a contract to autonomously terminate itself. When invoked, this opcode deallocates the contract, freeing up storage and computational resources on the Ethereum blockchain. Notably, the remaining Ether in the contract is sent to a specified address, ensuring a responsible handling of funds.



The contract is not vulnerable to ERC-20 approve Race condition vulnerability.

The ERC-20 race condition arises when two or more transactions attempt to interact with the same ERC-20 token contract concurrently. This scenario can result in conflicts and unexpected behavior due to the non-atomic nature of certain operations in the contract. Atomicity refers to the concept that an operation is indivisible and occurs as a single, uninterrupted unit.



The contract's owner was not found.

Renounced ownership indicates that the contract is truly decentralized, as the owner has relinquished control, ensuring that the contract's functionality and rules cannot be altered by administrators or any central authority.




No addresses contain more than 5% of circulating token supply.

Users with token balances exceeding 5% of the circulating token supply are critical to monitor, as their actions can significantly influence the token's price and ecosystem. Proper token distribution helps maintain a healthy market by preventing concentration of power and promoting fair participation.



The contracts are not using functions that can only be called by the owners. [Ownership Renounced]

An overpowered owner risk occurs when a contract has numerous functions that only the owner can execute. This can lead to centralization issues and potential abuse, as the owner has disproportionate control over the contract's operations.



The contract does not have a cooldown feature.

Cooldown functions, a crucial aspect in the smart contract landscape, are employed to temporarily suspend trading activities or other contract workflows. The mechanism introduces a time-based delay, effectively preventing users from repeatedly executing transactions or engaging in rapid buying and selling of tokens. Cooldown functions are used to halt trading or other contract workflows for a certain amount of time so as to prevent users from repeatedly executing transactions or buying and selling tokens.



Owners cannot whitelist tokens or users. [Ownership Renounced]

This empowers the contract owner to selectively grant privileges to users, such as exemption from fees or access to unique contract features.




Owners cannot set or update Fees in the contract. [Ownership Renounced]

In the context of smart contracts, fees are essential components that may be associated with various functionalities, such as transactions, token transfers, or other specific actions. The ability for owners to set or update fees is particularly valuable in scenarios where fee adjustments are needed to align with market conditions, regulatory requirements, or project-specific considerations. The Owners Can Set or Update Fees module focuses on identifying the capability within a smart contract for owners to establish or modify fees. This feature allows contract owners to have control over the fee structure within the contract, providing flexibility and adaptability to changing circumstances.




Hardcoded addresses were found.

The inclusion of a fixed or hardcoded address within a smart contract has the potential to pose significant challenges in the future, particularly concerning the contract's adaptability and upgradability. This static reference to an address may impede the seamless implementation of updates or modifications to the contract, hindering its ability to evolve in response to changing requirements. Such rigidity may result in complications and obstacles when attempting to enhance or alter the smart contract's functionality over time.



The contract does not have any owner-controlled functions modifying token balances. [Ownership Renounced]

The Owners Updating Token Balance module is focused on identifying situations where a smart contract has functions controlled by owners that allow them to update token balances for other users or the contract. If a contract permits owners to manipulate token balances, it can have significant implications on user holdings and overall contract integrity. In some scenarios, contracts may provide owners with functions that enable the manual adjustment of token balances. While this feature can be legitimate for specific use cases, such as token distribution or rewards, it also introduces potential risks. Allowing owners to arbitrarily update token balances may lead to vulnerabilities, manipulation, or unintended changes in the token ecosystem.



Owner's wallet contains 0 tokens which is less than 5% of the circulating token supply. [Ownership Renounced]

A check on the owner's wallet balance exceeding a specific token amount can indicate a centralization risk, where the owner may have disproportionate control over the token supply, potentially leading to manipulation or abuse.



No such functions retrieving ownership were found.

The Function Retrieving Ownership module serves the purpose of swiftly and efficiently retrieving ownership-related information within a smart contract. This functionality is vital for projects seeking to access and manage ownership data seamlessly. Utilizing this module, developers can streamline the process of obtaining ownership details, contributing to the effective administration of ownership-related functions within the ecosystem.



IS SPAM CONTRACT

A spam NFT is an NFT that is considered low-quality or deceptive, cluttering the marketplace and potentially misleading users.



Absence of Malicious Typecasting.

Malicious typecasting, particularly the conversion of uint160 values to addresses, is a tactic often used by scammers to create deceptive addresses that can bypass standard detection mechanisms, facilitating fraudulent activities.

LIQUIDITY BURN STATUS

The liquidity burn status indicates whether the LP tokens for the scanned contract have been permanently removed or remain accessible. If burnt, the LP tokens are sent to an irrecoverable address, ensuring that the liquidity cannot be withdrawn, offering permanent stability and security to the project. If not burnt, the LP tokens could still be accessed and withdrawn, which might expose investors to risks of liquidity manipulation or removal.

LIQUIDITY LOCK STATUS

The liquidity status determines whether the liquidity for the scanned contract is securely locked or accessible. If locked, LP tokens are stored in a time-locked contract, preventing any withdrawals until the lock expires. This helps protect investors from sudden liquidity removal. If not locked, LP tokens remain accessible, allowing project developers or liquidity providers to withdraw liquidity at any time, potentially posing risks to investors.

No such functions having totalSupply function update were found.

A fixed supply token is critical when the token's value is tied to scarcity or when precise control over inflation or deflation is required. Without a fixed supply, the contract could introduce unexpected inflation, devalue the token, or erode trust in the token's consistency.

No such functions having gas abuse via malicious minting.

Gas abuse refers to patterns within smart contracts that manipulate gas consumption in ways that unnecessarily increase transaction costs for users. This can occur through various mechanisms designed to exploit gas inefficiencies or inflate gas usage, shifting the financial burden onto users without their knowledge.

Valid token name or symbol.

The token name or symbol contains potentially harmful content, such as HTML tags or JavaScript code. If these unsanitized strings are displayed by user interfaces, they could execute scripts in users' browsers, posing a significant risk of Cross-Site Scripting (XSS).

 **No such functions having addresses with special access.**


Special permissions granted to non-owner addresses allow them to execute specific functions with elevated access. This can introduce security risks, as these privileged addresses may perform critical operations that impact the contract's state or user funds. If not properly managed or monitored, these permissions could lead to unauthorized or malicious actions, compromising the contract's integrity.

 **No hidden owner detected**

The Hidden Owner check identifies whether there are any hidden owner roles within the contract. Hidden ownership can allow unauthorized access and control over contract functions, which poses a risk to users and stakeholders.

 **The token is not a counterfeit token**

The contract is found to have the token symbol identical to that of official tokens, thereby falling under the category of counterfeit tokens. These counterfeit tokens can mislead users into believing they are interacting with legitimate, well-known cryptocurrencies, potentially leading to financial losses and damaging the reputation of the official token.

 **Absence of external call risk in critical functions.**

This check identifies risks associated with external calls within critical functions. External calls can introduce vulnerabilities such as unexpected state changes, or dependencies on external contracts, which may compromise the integrity and reliability of the function's execution.

 **The contract is not a honeypot.**

A token is honeypot when it exhibits the typical characteristics designed to trap investors' funds while preventing them from profiting. Common signs include the inability to sell tokens once purchased, restrictive transfer functions coded into the smart contract, excessive or hidden taxes on transactions, or permissions that allow only the contract owner to execute sells.

Issue Type

Action Taken

SOLIDITY PRAGMA VERSION

 Pending Fix

Description

The contract should be written using the latest Solidity pragma version as it comes with numerous bug fixes. Utilizing an outdated version exposes the contract to vulnerabilities associated with known issues that have been addressed in subsequent updates. Therefore, it is essential to stay current with the latest Solidity version to ensure the robustness and security of the contract against potential vulnerabilities.

Remediation

Update the Solidity pragma version to the latest stable version to benefit from the latest bug fixes and security enhancements.

File Location

Line No.


CompoundX_Mainnet.sol 

L2 - L2

Issue Type

Action Taken

HARDCODED ADDRESSES

 Pending Fix

Description

The inclusion of a fixed or hardcoded address within a smart contract has the potential to pose significant challenges in the future, particularly concerning the contract's adaptability and upgradability. This static reference to an address may impede the seamless implementation of updates or modifications to the contract, hindering its ability to evolve in response to changing requirements. Such rigidity may result in complications and obstacles when attempting to enhance or alter the smart contract's functionality over time.

Remediation

To remediate, consider using configuration variables or function parameters instead of hardcoded addresses to allow for flexibility and easier updates in the future. If a fixed address is necessary, ensure that it is clearly documented and easily modifiable in the contract.

File Location

Line No.

CompoundX_Mainnet.sol 

L127 - L127

CompoundX_Mainnet.sol 

L127 - L127

CompoundX_Mainnet.sol 

L142 - L142

04. Findings Summary



0x8da44609d55ae3372c672d66be0e12776c609731

BINANCE (Bsc Mainnet) | [View on Bscscan](#)



Security Score

98.56/100



Scan duration

109 secs



Lines of code

834



0

Crit

0

High

0

Med

0

Low

293

Info

181

Gas



This audit report has been verified by the SolidityScan team. To learn more about our published reports. [click here](#)

ACTION TAKEN

<p style="font-size: 24px; font-weight: bold;">0</p> <p> Fixed</p>	<p style="font-size: 24px; font-weight: bold;">1</p> <p> False Positive</p>	<p style="font-size: 24px; font-weight: bold;">475</p> <p> Won't Fix</p>	<p style="font-size: 24px; font-weight: bold;">122</p> <p> Pending Fix</p>
---	--	---	---

S. No.	Severity	Bug Type	Instances	Detection Method	Status
H001	● High	ADMIN FEE CHARGED WITHOUT REDUCING CREDITED PRINCIPAL CAUSES SYSTEMIC UNDER-COLLATERALIZATION	1	SolidityScan AI	False Positive
M001	● Medium	REWARDS CAN BE RETROACTIVELY BOOSTED BY CURRENT ROI TIER WHEN AUTO-CALCULATION IS SKIPPED	1	SolidityScan AI	Pending Fix
I001	● Informational	ADDING A RETURN STATEMENT WHEN THE FUNCTION DEFINES A NAMED RETURN VARIABLE IS REDUNDANT	7	Automated	Pending Fix
I002	● Informational	HARD-CODED ADDRESS DETECTED	2	Automated	Pending Fix
I003	● Informational	BLOCK VALUES AS A PROXY FOR TIME	31	Automated	Pending Fix
I004	● Informational	CONSIDER USING UINT48 FOR TIME-RELATED VARIABLES	16	Automated	Pending Fix
I005	● Informational	CONSTRUCTORS SHOULD EMIT AN EVENT	1	Automated	Pending Fix
I006	● Informational	IF-STATEMENT REFACTORING	3	Automated	Pending Fix
I007	● Informational	MISSING @AUTHOR IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS	1	Automated	Pending Fix
I008	● Informational	MISSING @AUTHOR IN NATSPEC COMMENTS FOR CONTRACT DECLARATION	2	Automated	Pending Fix
I009	● Informational	MISSING @DEV IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS	1	Automated	Pending Fix
I010	● Informational	MISSING @DEV IN NATSPEC COMMENTS FOR CONTRACT DECLARATION	2	Automated	Pending Fix

S. No.	Severity	Bug Type	Instances	Detection Method	Status
I011	● Informational	MISSING @DEV IN NATSPEC COMMENTS FOR FUNCTIONS	42	Automated	⚠️ <i>Pending Fix</i>
I012	● Informational	MISSING @INHERITDOC ON OVERRIDE FUNCTIONS	44	Automated	⚠️ <i>Pending Fix</i>
I013	● Informational	MISSING NATSPEC COMMENTS IN SCOPE BLOCKS	69	Automated	⚠️ <i>Pending Fix</i>
I014	● Informational	MISSING NATSPEC DESCRIPTIONS FOR PUBLIC VARIABLE DECLARATIONS	38	Automated	⚠️ <i>Pending Fix</i>
I015	● Informational	MISSING @NOTICE IN NATSPEC COMMENTS FOR CONSTRUCTORS	2	Automated	⚠️ <i>Pending Fix</i>
I016	● Informational	MISSING @NOTICE IN NATSPEC COMMENTS FOR FUNCTIONS	42	Automated	⚠️ <i>Pending Fix</i>
I017	● Informational	MISSING @NOTICE IN NATSPEC COMMENTS FOR MODIFIERS	1	Automated	⚠️ <i>Pending Fix</i>
I018	● Informational	MISSING UNDERSCORE IN NAMING VARIABLES	1	Automated	⚠️ <i>Pending Fix</i>
I019	● Informational	NAME MAPPING PARAMETERS	6	Automated	⚠️ <i>Pending Fix</i>
I020	● Informational	VARIABLES SHOULD BE IMMUTABLE	1	Automated	⚠️ <i>Pending Fix</i>
G001	● Gas	ARRAY LENGTH CACHING	7	Automated	⚠️ <i>Pending Fix</i>
G002	● Gas	ASSIGNING TO STRUCTS CAN BE MORE EFFICIENT	3	Automated	⚠️ <i>Pending Fix</i>
G003	● Gas	AVOID ZERO-TO-ONE STORAGE WRITES	1	Automated	⚠️ <i>Pending Fix</i>
G004	● Gas	CHEAPER CONDITIONAL OPERATORS	17	Automated	⚠️ <i>Pending Fix</i>
G005	● Gas	CHEAPER INEQUALITIES IN IF()	25	Automated	⚠️ <i>Pending Fix</i>
G006	● Gas	CHEAPER INEQUALITIES IN REQUIRE()	19	Automated	⚠️ <i>Pending Fix</i>
G007	● Gas	DEFAULT INT VALUES ARE MANUALLY RESET	9	Automated	⚠️ <i>Pending Fix</i>

S. No.	Severity	Bug Type	Instances	Detection Method	Status
G008	● Gas	DEFINE CONSTRUCTOR AS PAYABLE	1	Automated	⚠️ <i>Pending Fix</i>
G009	● Gas	EMIT USED IN LOOP	2	Automated	⚠️ <i>Pending Fix</i>
G010	● Gas	FUNCTIONS CAN BE IN-LINED	7	Automated	⚠️ <i>Pending Fix</i>
G011	● Gas	FUNCTION SHOULD RETURN STRUCT	6	Automated	⚠️ <i>Pending Fix</i>
G012	● Gas	GAS INEFFICIENCY DUE TO MULTIPLE OPERANDS IN SINGLE IF/ELSEIF CONDITION	9	Automated	⚠️ <i>Pending Fix</i>
G013	● Gas	GAS OPTIMIZATION FOR STATE VARIABLES	3	Automated	⚠️ <i>Pending Fix</i>
G014	● Gas	GAS OPTIMIZATION IN DECREMENTS	3	Automated	⚠️ <i>Pending Fix</i>
G015	● Gas	GAS OPTIMIZATION IN INCREMENTS	7	Automated	⚠️ <i>Pending Fix</i>
G016	● Gas	LONG NUMBER LITERALS	1	Automated	⚠️ <i>Pending Fix</i>
G017	● Gas	LONG REQUIRE/REVERT STRINGS	1	Automated	⚠️ <i>Pending Fix</i>
G018	● Gas	NAMED RETURN OF LOCAL VARIABLE SAVES GAS AS COMPARED TO RETURN STATEMENT	1	Automated	⚠️ <i>Pending Fix</i>
G019	● Gas	PUBLIC CONSTANTS CAN BE PRIVATE	20	Automated	⚠️ <i>Pending Fix</i>
G020	● Gas	SMALLER DATA TYPES COST MORE	11	Automated	⚠️ <i>Pending Fix</i>
G021	● Gas	SPLITTING REQUIRE STATEMENTS	7	Automated	⚠️ <i>Pending Fix</i>
G022	● Gas	STORAGE VARIABLE CACHING IN MEMORY	42	Automated	⚠️ <i>Pending Fix</i>
G023	● Gas	SUPERFLUOUS EVENT FIELDS	11	Automated	⚠️ <i>Pending Fix</i>
G024	● Gas	UNNECESSARY CHECKED ARITHMETIC IN LOOP	11	Automated	⚠️ <i>Pending Fix</i>
G025	● Gas	UNNECESSARY DEFAULT VALUE INITIALIZATION	1	Automated	⚠️ <i>Pending Fix</i>

05. Vulnerability Details

Issue Type

ADMIN FEE CHARGED WITHOUT REDUCING CREDITED PRINCIPAL CAUSES SYSTEMIC UNDER-COLLATERALIZATION

S. No.	Severity	Detection Method	Instances
H001	● High	 SolidityScan AI	1

Description

On `deposit`, `compoundDailyReward`, and `compoundNetworkReward`, the contract transfers an admin fee out but credits the full `amount` as a new position. This means liabilities (positions + reward accruals) are based on the gross amount, while the contract only retains the net amount after the fee. Over time, this creates a shortfall between contract assets and owed rewards.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_529	CompoundX_Mainnet.sol 	L184 - L234	 <i>False Positive</i>


Issue Type

REWARDS CAN BE RETROACTIVELY BOOSTED BY CURRENT ROI TIER WHEN AUTO-CALCULATION IS SKIPPED

S. No.	Severity	Detection Method	Instances
M001	● Medium	✦✦ SolidityScan AI	1

Description

The reward engine computes a single `roiTier` based on the current `users[userAddr].totalActiveDeposit` and applies it to all positions for the entire elapsed period (`_calculateAndUpdateDailyRewards`). When `_shouldSkipAutoCalculation` returns true (e.g., users with \geq MIN_POSITIONS_FOR_MANUAL and within the manual interval), deposits can be made without updating checkpoints. A subsequent manual calculation uses the new higher tier for time that accrued before the deposit, inflating rewards retroactively.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_528	CompoundX_Mainnet.sol ↗	L962 - L1006	 Won't Fix

Issue Type

ADDING A RETURN STATEMENT WHEN THE FUNCTION DEFINES A NAMED RETURN VARIABLE IS REDUNDANT

S. No.	Severity	Detection Method	Instances
I001	● Informational	Automated	7

Description

The contract defines a named return despite having a return statement. This causes redundancy and can lead to potential confusion in the code.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_77	CompoundX_Mainnet.sol ↗	L498 - L521	 <i>Won't Fix</i>
SSB_5650526_78	CompoundX_Mainnet.sol ↗	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_79	CompoundX_Mainnet.sol ↗	L607 - L624	 <i>Won't Fix</i>
SSB_5650526_80	CompoundX_Mainnet.sol ↗	L626 - L643	 <i>Won't Fix</i>
SSB_5650526_81	CompoundX_Mainnet.sol ↗	L671 - L696	 <i>Won't Fix</i>
SSB_5650526_82	CompoundX_Mainnet.sol ↗	L698 - L701	 <i>Won't Fix</i>
SSB_5650526_83	CompoundX_Mainnet.sol ↗	L719 - L747	 <i>Won't Fix</i>

Issue Type

HARD-CODED ADDRESS DETECTED

S. No.	Severity	Detection Method	Instances
I002	● Informational	Automated	2





Description

The contract contains an unknown hard-coded address. This address might be used for some malicious activity. Please check the hard-coded address and its usage.

These hard-coded addresses may be used everywhere throughout the code to define states and interact with the functions and external calls.

Therefore, it is extremely crucial to ensure the correctness of these token contracts as they define various important aspects of the protocol operation.

A misconfigured address mapping could lead to the potential loss of user funds or compromise of the contract owner depending on the function logic.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_389	CompoundX_Mainnet.sol 	L127 - L127	 Won't Fix
SSB_5650526_390	CompoundX_Mainnet.sol 	L142 - L142	 Won't Fix

Issue Type

BLOCK VALUES AS A PROXY FOR TIME





S. No.	Severity	Detection Method	Instances
I003	● Informational	Automated	31



















Description














Contracts often need access to time values to perform certain types of functionality. Values such as `block.timestamp` and `block.number` can be used to determine the current time or the time delta. However, they are not recommended for most use cases.

For `block.number`, as Ethereum block times are generally around 14 seconds, the delta between blocks can be predicted. The block times, on the other hand, do not remain constant and are subject to change for a number of reasons, e.g., fork reorganizations and the difficulty bomb.

Due to variable block times, `block.number` should not be relied on for precise calculations of time.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_265	CompoundX_Mainnet.sol ↗	L156 - L156	 <i>Won't Fix</i>
SSB_5650526_266	CompoundX_Mainnet.sol ↗	L185 - L185	 <i>Won't Fix</i>
SSB_5650526_267	CompoundX_Mainnet.sol ↗	L212 - L212	 <i>Won't Fix</i>
SSB_5650526_267	CompoundX_Mainnet.sol ↗	L328 - L328	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_267	CompoundX_Mainnet.sol ↗	L416 - L416	 <i>Won't Fix</i>
SSB_5650526_268	CompoundX_Mainnet.sol ↗	L213 - L213	 <i>Won't Fix</i>
SSB_5650526_268	CompoundX_Mainnet.sol ↗	L329 - L329	 <i>Won't Fix</i>
SSB_5650526_268	CompoundX_Mainnet.sol ↗	L417 - L417	 <i>Won't Fix</i>
SSB_5650526_269	CompoundX_Mainnet.sol ↗	L237 - L237	 <i>Won't Fix</i>
SSB_5650526_269	CompoundX_Mainnet.sol ↗	L287 - L287	 <i>Won't Fix</i>
SSB_5650526_269	CompoundX_Mainnet.sol ↗	L386 - L386	 <i>Won't Fix</i>
SSB_5650526_270	CompoundX_Mainnet.sol ↗	L346 - L346	 <i>Won't Fix</i>
SSB_5650526_271	CompoundX_Mainnet.sol ↗	L448 - L448	 <i>Won't Fix</i>
SSB_5650526_272	CompoundX_Mainnet.sol ↗	L449 - L449	 <i>Won't Fix</i>
SSB_5650526_272	CompoundX_Mainnet.sol ↗	L928 - L928	 <i>Won't Fix</i>
SSB_5650526_272	CompoundX_Mainnet.sol ↗	L969 - L969	 <i>Won't Fix</i>
SSB_5650526_272	CompoundX_Mainnet.sol ↗	L1015 - L1015	 <i>Won't Fix</i>
SSB_5650526_273	CompoundX_Mainnet.sol ↗	L456 - L456	 <i>Won't Fix</i>
SSB_5650526_274	CompoundX_Mainnet.sol ↗	L464 - L464	 <i>Won't Fix</i>
SSB_5650526_275	CompoundX_Mainnet.sol ↗	L478 - L478	 <i>Won't Fix</i>
SSB_5650526_276	CompoundX_Mainnet.sol ↗	L657 - L657	 <i>Won't Fix</i>
SSB_5650526_277	CompoundX_Mainnet.sol ↗	L694 - L694	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_278	CompoundX_Mainnet.sol 	L731 - L731	 <i>Won't Fix</i>
SSB_5650526_279	CompoundX_Mainnet.sol 	L732 - L732	 <i>Won't Fix</i>
SSB_5650526_280	CompoundX_Mainnet.sol 	L925 - L925	 <i>Won't Fix</i>
SSB_5650526_281	CompoundX_Mainnet.sol 	L932 - L932	 <i>Won't Fix</i>
SSB_5650526_282	CompoundX_Mainnet.sol 	L941 - L941	 <i>Won't Fix</i>
SSB_5650526_283	CompoundX_Mainnet.sol 	L945 - L945	 <i>Won't Fix</i>
SSB_5650526_284	CompoundX_Mainnet.sol 	L948 - L948	 <i>Won't Fix</i>
SSB_5650526_285	CompoundX_Mainnet.sol 	L963 - L963	 <i>Won't Fix</i>
SSB_5650526_285	CompoundX_Mainnet.sol 	L1009 - L1009	 <i>Won't Fix</i>


Issue Type

CONSIDER USING UINT48 FOR TIME-RELATED VARIABLES

S. No.	Severity	Detection Method	Instances
I004	● Informational	Automated	16

Description

The contract defines time-related variables as `uint256`. Since Unix timestamps can be adequately represented using `uint48`, utilizing `uint256` for these variables results in unnecessary gas consumption due to larger storage requirements.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_417	CompoundX_Mainnet.sol ↗	L45 - L45	 <i>Won't Fix</i>
SSB_5650526_418	CompoundX_Mainnet.sol ↗	L449 - L449	 <i>Won't Fix</i>
SSB_5650526_418	CompoundX_Mainnet.sol ↗	L928 - L928	 <i>Won't Fix</i>
SSB_5650526_418	CompoundX_Mainnet.sol ↗	L969 - L969	 <i>Won't Fix</i>
SSB_5650526_418	CompoundX_Mainnet.sol ↗	L1015 - L1015	 <i>Won't Fix</i>
SSB_5650526_419	CompoundX_Mainnet.sol ↗	L461 - L461	 <i>Won't Fix</i>
SSB_5650526_420	CompoundX_Mainnet.sol ↗	L557 - L557	 <i>Won't Fix</i>
SSB_5650526_421	CompoundX_Mainnet.sol ↗	L656 - L656	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_422	CompoundX_Mainnet.sol 	L663 - L663	 <i>Won't Fix</i>
SSB_5650526_423	CompoundX_Mainnet.sol 	L732 - L732	 <i>Won't Fix</i>
SSB_5650526_424	CompoundX_Mainnet.sol 	L925 - L925	 <i>Won't Fix</i>
SSB_5650526_425	CompoundX_Mainnet.sol 	L932 - L932	 <i>Won't Fix</i>
SSB_5650526_426	CompoundX_Mainnet.sol 	L933 - L933	 <i>Won't Fix</i>
SSB_5650526_427	CompoundX_Mainnet.sol 	L945 - L945	 <i>Won't Fix</i>
SSB_5650526_428	CompoundX_Mainnet.sol 	L986 - L986	 <i>Won't Fix</i>
SSB_5650526_428	CompoundX_Mainnet.sol 	L1032 - L1032	 <i>Won't Fix</i>


Issue Type

CONSTRUCTORS SHOULD EMIT AN EVENT

S. No.	Severity	Detection Method	Instances
I005	● Informational	Automated	1

Description

A constructor in a smart contract executes only once during deployment, making it crucial to log this action for transparency. Without an emitted event, tracking contract deployment becomes difficult, reducing the ability to audit and monitor contract instances.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_199	CompoundX_Mainnet.sol ↗	L12 - L14	 Won't Fix




Issue Type

IF-STATEMENT REFACTORING

S. No.	Severity	Detection Method	Instances
I006	● Informational	Automated	3

Description

In Solidity, we aim to write clear, efficient code that is both easy to understand and maintain. If statements can be converted to ternary operators. While using ternary operators instead of if/else statements can sometimes lead to more concise code, it's crucial to understand the trade-offs involved.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_386	CompoundX_Mainnet.sol ↗	L854 - L866	 <i>Won't Fix</i>
SSB_5650526_387	CompoundX_Mainnet.sol ↗	L875 - L879	 <i>Won't Fix</i>
SSB_5650526_388	CompoundX_Mainnet.sol ↗	L997 - L1001	 <i>Won't Fix</i>

Issue Type

MISSING @AUTHOR IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS

S. No.	Severity	Detection Method	Instances
I007	● Informational	Automated	1



Description

Natspec `@author` tags are missing from abstract contract declarations in the code. This reduces transparency and makes it difficult to track contract authorship, which is important for accountability, maintenance, and reference purposes.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_200	CompoundX_Mainnet.sol ↗	L7 - L29	<i>Won't Fix</i>



Issue Type

MISSING @AUTHOR IN NATSPEC COMMENTS FOR CONTRACT DECLARATION

S. No.	Severity	Detection Method	Instances
I008	● Informational	Automated	2

Description

Natspec `@author` tags are missing from contract declarations in the code. This reduces code clarity and makes it difficult to determine the original author or organization responsible for the contract. The absence of an `@author` tag can lead to challenges in tracking ownership, verifying authenticity, and maintaining proper documentation.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_384	CompoundX_Mainnet.sol ↗	L7 - L29	 <i>Won't Fix</i>
SSB_5650526_385	CompoundX_Mainnet.sol ↗	L31 - L1054	 <i>Won't Fix</i>

Issue Type

MISSING @DEV IN NATSPEC COMMENTS FOR ABSTRACT CONTRACTS

S. No.	Severity	Detection Method	Instances
I009	● Informational	Automated	1



Description

Natspec `@dev` tags are missing from abstract contract declarations in the code. This reduces clarity for developers using or inheriting these contracts, making it harder to understand their intended use, design considerations, and any important implementation details.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_526	CompoundX_Mainnet.sol ↗	L7 - L29	<i>Won't Fix</i>

Issue Type

MISSING @DEV IN NATSPEC COMMENTS FOR CONTRACT DECLARATION

S. No.	Severity	Detection Method	Instances
I010	● Informational	Automated	2



Description

Natspec `@dev` tags are missing from contract declarations in the code. This reduces code clarity and makes it difficult for developers to understand the contract's internal logic, implementation details, and potential caveats.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_382	CompoundX_Mainnet.sol ↗	L7 - L29	<i>Won't Fix</i>
SSB_5650526_383	CompoundX_Mainnet.sol ↗	L31 - L1054	<i>Won't Fix</i>

Issue Type





































MISSING @DEV IN NATSPEC COMMENTS FOR FUNCTIONS

































S. No.	Severity	Detection Method	Instances
I011	● Informational	Automated	42

Description

Natspec `@dev` comments are missing from a function declaration in the code. The `@dev` tag provides a detailed explanation of a function's purpose, assumptions, and behavior. Lack of `@dev` documentation can: reduce code maintainability, making it harder for developers to understand the function's implementation details.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_475	CompoundX_Mainnet.sol ↗	L16 - L18	 <i>Won't Fix</i>
SSB_5650526_476	CompoundX_Mainnet.sol ↗	L25 - L28	 <i>Won't Fix</i>
SSB_5650526_477	CompoundX_Mainnet.sol ↗	L155 - L182	 <i>Won't Fix</i>
SSB_5650526_478	CompoundX_Mainnet.sol ↗	L184 - L234	 <i>Won't Fix</i>
SSB_5650526_479	CompoundX_Mainnet.sol ↗	L236 - L284	 <i>Won't Fix</i>
SSB_5650526_480	CompoundX_Mainnet.sol ↗	L286 - L343	 <i>Won't Fix</i>
SSB_5650526_481	CompoundX_Mainnet.sol ↗	L345 - L383	 <i>Won't Fix</i>
SSB_5650526_482	CompoundX_Mainnet.sol ↗	L385 - L431	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_483	CompoundX_Mainnet.sol 	L433 - L438	 <i>Won't Fix</i>
SSB_5650526_484	CompoundX_Mainnet.sol 	L440 - L445	 <i>Won't Fix</i>
SSB_5650526_485	CompoundX_Mainnet.sol 	L447 - L452	 <i>Won't Fix</i>
SSB_5650526_486	CompoundX_Mainnet.sol 	L454 - L459	 <i>Won't Fix</i>
SSB_5650526_487	CompoundX_Mainnet.sol 	L461 - L471	 <i>Won't Fix</i>
SSB_5650526_488	CompoundX_Mainnet.sol 	L473 - L483	 <i>Won't Fix</i>
SSB_5650526_489	CompoundX_Mainnet.sol 	L485 - L496	 <i>Won't Fix</i>
SSB_5650526_490	CompoundX_Mainnet.sol 	L498 - L521	 <i>Won't Fix</i>
SSB_5650526_491	CompoundX_Mainnet.sol 	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_492	CompoundX_Mainnet.sol 	L555 - L567	 <i>Won't Fix</i>
SSB_5650526_493	CompoundX_Mainnet.sol 	L569 - L593	 <i>Won't Fix</i>
SSB_5650526_494	CompoundX_Mainnet.sol 	L595 - L605	 <i>Won't Fix</i>
SSB_5650526_495	CompoundX_Mainnet.sol 	L607 - L624	 <i>Won't Fix</i>
SSB_5650526_496	CompoundX_Mainnet.sol 	L626 - L643	 <i>Won't Fix</i>
SSB_5650526_497	CompoundX_Mainnet.sol 	L645 - L647	 <i>Won't Fix</i>
SSB_5650526_498	CompoundX_Mainnet.sol 	L649 - L651	 <i>Won't Fix</i>
SSB_5650526_499	CompoundX_Mainnet.sol 	L653 - L669	 <i>Won't Fix</i>
SSB_5650526_500	CompoundX_Mainnet.sol 	L671 - L696	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_501	CompoundX_Mainnet.sol 	L698 - L701	 <i>Won't Fix</i>
SSB_5650526_502	CompoundX_Mainnet.sol 	L704 - L706	 <i>Won't Fix</i>
SSB_5650526_503	CompoundX_Mainnet.sol 	L708 - L717	 <i>Won't Fix</i>
SSB_5650526_504	CompoundX_Mainnet.sol 	L719 - L747	 <i>Won't Fix</i>
SSB_5650526_505	CompoundX_Mainnet.sol 	L749 - L776	 <i>Won't Fix</i>
SSB_5650526_506	CompoundX_Mainnet.sol 	L778 - L793	 <i>Won't Fix</i>
SSB_5650526_507	CompoundX_Mainnet.sol 	L795 - L827	 <i>Won't Fix</i>
SSB_5650526_508	CompoundX_Mainnet.sol 	L829 - L845	 <i>Won't Fix</i>
SSB_5650526_509	CompoundX_Mainnet.sol 	L847 - L887	 <i>Won't Fix</i>
SSB_5650526_510	CompoundX_Mainnet.sol 	L889 - L907	 <i>Won't Fix</i>
SSB_5650526_511	CompoundX_Mainnet.sol 	L909 - L917	 <i>Won't Fix</i>
SSB_5650526_512	CompoundX_Mainnet.sol 	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_513	CompoundX_Mainnet.sol 	L940 - L960	 <i>Won't Fix</i>
SSB_5650526_514	CompoundX_Mainnet.sol 	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_515	CompoundX_Mainnet.sol 	L1008 - L1037	 <i>Won't Fix</i>
SSB_5650526_516	CompoundX_Mainnet.sol 	L1039 - L1053	 <i>Won't Fix</i>




Issue Type





































MISSING @INHERITDOC ON OVERRIDE FUNCTIONS



















S. No.	Severity	Detection Method	Instances
I012	● Informational	Automated	44







Description

Overridden functions in Solidity should include the `@inheritdoc` tag in their NatSpec comments to ensure documentation consistency and clarity. Failing to include `@inheritdoc` can lead to incomplete or missing documentation, making it difficult for developers and auditors to understand the function's intended behavior. This lack of clarity can increase maintenance overhead and potentially lead to misinterpretations of contract logic.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_430	CompoundX_Mainnet.sol ↗	L12 - L14	 <i>Won't Fix</i>
SSB_5650526_431	CompoundX_Mainnet.sol ↗	L16 - L18	 <i>Won't Fix</i>
SSB_5650526_432	CompoundX_Mainnet.sol ↗	L25 - L28	 <i>Won't Fix</i>
SSB_5650526_433	CompoundX_Mainnet.sol ↗	L141 - L153	 <i>Won't Fix</i>
SSB_5650526_434	CompoundX_Mainnet.sol ↗	L155 - L182	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_435	CompoundX_Mainnet.sol 	L184 - L234	 <i>Won't Fix</i>
SSB_5650526_436	CompoundX_Mainnet.sol 	L236 - L284	 <i>Won't Fix</i>
SSB_5650526_437	CompoundX_Mainnet.sol 	L286 - L343	 <i>Won't Fix</i>
SSB_5650526_438	CompoundX_Mainnet.sol 	L345 - L383	 <i>Won't Fix</i>
SSB_5650526_439	CompoundX_Mainnet.sol 	L385 - L431	 <i>Won't Fix</i>
SSB_5650526_440	CompoundX_Mainnet.sol 	L433 - L438	 <i>Won't Fix</i>
SSB_5650526_441	CompoundX_Mainnet.sol 	L440 - L445	 <i>Won't Fix</i>
SSB_5650526_442	CompoundX_Mainnet.sol 	L447 - L452	 <i>Won't Fix</i>
SSB_5650526_443	CompoundX_Mainnet.sol 	L454 - L459	 <i>Won't Fix</i>
SSB_5650526_444	CompoundX_Mainnet.sol 	L461 - L471	 <i>Won't Fix</i>
SSB_5650526_445	CompoundX_Mainnet.sol 	L473 - L483	 <i>Won't Fix</i>
SSB_5650526_446	CompoundX_Mainnet.sol 	L485 - L496	 <i>Won't Fix</i>
SSB_5650526_447	CompoundX_Mainnet.sol 	L498 - L521	 <i>Won't Fix</i>
SSB_5650526_448	CompoundX_Mainnet.sol 	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_449	CompoundX_Mainnet.sol 	L555 - L567	 <i>Won't Fix</i>
SSB_5650526_450	CompoundX_Mainnet.sol 	L569 - L593	 <i>Won't Fix</i>
SSB_5650526_451	CompoundX_Mainnet.sol 	L595 - L605	 <i>Won't Fix</i>
SSB_5650526_452	CompoundX_Mainnet.sol 	L607 - L624	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_453	CompoundX_Mainnet.sol ↗	L626 - L643	 <i>Won't Fix</i>
SSB_5650526_454	CompoundX_Mainnet.sol ↗	L645 - L647	 <i>Won't Fix</i>
SSB_5650526_455	CompoundX_Mainnet.sol ↗	L649 - L651	 <i>Won't Fix</i>
SSB_5650526_456	CompoundX_Mainnet.sol ↗	L653 - L669	 <i>Won't Fix</i>
SSB_5650526_457	CompoundX_Mainnet.sol ↗	L671 - L696	 <i>Won't Fix</i>
SSB_5650526_458	CompoundX_Mainnet.sol ↗	L698 - L701	 <i>Won't Fix</i>
SSB_5650526_459	CompoundX_Mainnet.sol ↗	L704 - L706	 <i>Won't Fix</i>
SSB_5650526_460	CompoundX_Mainnet.sol ↗	L708 - L717	 <i>Won't Fix</i>
SSB_5650526_461	CompoundX_Mainnet.sol ↗	L719 - L747	 <i>Won't Fix</i>
SSB_5650526_462	CompoundX_Mainnet.sol ↗	L749 - L776	 <i>Won't Fix</i>
SSB_5650526_463	CompoundX_Mainnet.sol ↗	L778 - L793	 <i>Won't Fix</i>
SSB_5650526_464	CompoundX_Mainnet.sol ↗	L795 - L827	 <i>Won't Fix</i>
SSB_5650526_465	CompoundX_Mainnet.sol ↗	L829 - L845	 <i>Won't Fix</i>
SSB_5650526_466	CompoundX_Mainnet.sol ↗	L847 - L887	 <i>Won't Fix</i>
SSB_5650526_467	CompoundX_Mainnet.sol ↗	L889 - L907	 <i>Won't Fix</i>
SSB_5650526_468	CompoundX_Mainnet.sol ↗	L909 - L917	 <i>Won't Fix</i>
SSB_5650526_469	CompoundX_Mainnet.sol ↗	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_470	CompoundX_Mainnet.sol ↗	L940 - L960	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_471	CompoundX_Mainnet.sol 	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_472	CompoundX_Mainnet.sol 	L1008 - L1037	 <i>Won't Fix</i>
SSB_5650526_473	CompoundX_Mainnet.sol 	L1039 - L1053	 <i>Won't Fix</i>

Issue Type





































MISSING NATSPEC COMMENTS IN SCOPE BLOCKS





































S. No.	Severity	Detection Method	Instances
I013	● Informational	Automated	69



















Description











NatSpec comments are missing from an unnamed scope block. Scope blocks can group computations or temporary logic and should be documented to clarify purpose and expected behavior. Lack of NatSpec documentation can reduce readability, making it hard to understand why the scope block exists.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_294	CompoundX_Mainnet.sol ↗	L164 - L169	 Won't Fix
SSB_5650526_295	CompoundX_Mainnet.sol ↗	L196 - L203	 Won't Fix
SSB_5650526_296	CompoundX_Mainnet.sol ↗	L198 - L202	 Won't Fix
SSB_5650526_297	CompoundX_Mainnet.sol ↗	L227 - L231	 Won't Fix
SSB_5650526_298	CompoundX_Mainnet.sol ↗	L248 - L251	 Won't Fix
SSB_5650526_298	CompoundX_Mainnet.sol ↗	L299 - L302	 Won't Fix
SSB_5650526_299	CompoundX_Mainnet.sol ↗	L259 - L261	 Won't Fix
SSB_5650526_299	CompoundX_Mainnet.sol ↗	L310 - L312	 Won't Fix
SSB_5650526_299	CompoundX_Mainnet.sol ↗	L358 - L360	 Won't Fix
SSB_5650526_299	CompoundX_Mainnet.sol ↗	L399 - L401	 Won't Fix

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_300	CompoundX_Mainnet.sol 	L267 - L269	 <i>Won't Fix</i>
SSB_5650526_301	CompoundX_Mainnet.sol 	L467 - L469	 <i>Won't Fix</i>
SSB_5650526_302	CompoundX_Mainnet.sol 	L476 - L482	 <i>Won't Fix</i>
SSB_5650526_303	CompoundX_Mainnet.sol 	L478 - L481	 <i>Won't Fix</i>
SSB_5650526_304	CompoundX_Mainnet.sol 	L488 - L494	 <i>Won't Fix</i>
SSB_5650526_305	CompoundX_Mainnet.sol 	L490 - L492	 <i>Won't Fix</i>
SSB_5650526_306	CompoundX_Mainnet.sol 	L559 - L564	 <i>Won't Fix</i>
SSB_5650526_307	CompoundX_Mainnet.sol 	L561 - L563	 <i>Won't Fix</i>
SSB_5650526_308	CompoundX_Mainnet.sol 	L572 - L574	 <i>Won't Fix</i>
SSB_5650526_309	CompoundX_Mainnet.sol 	L576 - L578	 <i>Won't Fix</i>
SSB_5650526_310	CompoundX_Mainnet.sol 	L580 - L582	 <i>Won't Fix</i>
SSB_5650526_311	CompoundX_Mainnet.sol 	L584 - L586	 <i>Won't Fix</i>
SSB_5650526_312	CompoundX_Mainnet.sol 	L588 - L590	 <i>Won't Fix</i>
SSB_5650526_313	CompoundX_Mainnet.sol 	L596 - L598	 <i>Won't Fix</i>
SSB_5650526_314	CompoundX_Mainnet.sol 	L600 - L602	 <i>Won't Fix</i>
SSB_5650526_315	CompoundX_Mainnet.sol 	L660 - L662	 <i>Won't Fix</i>
SSB_5650526_316	CompoundX_Mainnet.sol 	L662 - L666	 <i>Won't Fix</i>
SSB_5650526_317	CompoundX_Mainnet.sol 	L731 - L736	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_318	CompoundX_Mainnet.sol 	L752 - L775	 <i>Won't Fix</i>
SSB_5650526_319	CompoundX_Mainnet.sol 	L757 - L759	 <i>Won't Fix</i>
SSB_5650526_320	CompoundX_Mainnet.sol 	L759 - L761	 <i>Won't Fix</i>
SSB_5650526_321	CompoundX_Mainnet.sol 	L761 - L763	 <i>Won't Fix</i>
SSB_5650526_322	CompoundX_Mainnet.sol 	L763 - L765	 <i>Won't Fix</i>
SSB_5650526_323	CompoundX_Mainnet.sol 	L767 - L772	 <i>Won't Fix</i>
SSB_5650526_324	CompoundX_Mainnet.sol 	L783 - L785	 <i>Won't Fix</i>
SSB_5650526_325	CompoundX_Mainnet.sol 	L785 - L787	 <i>Won't Fix</i>
SSB_5650526_326	CompoundX_Mainnet.sol 	L789 - L792	 <i>Won't Fix</i>
SSB_5650526_327	CompoundX_Mainnet.sol 	L801 - L826	 <i>Won't Fix</i>
SSB_5650526_328	CompoundX_Mainnet.sol 	L806 - L811	 <i>Won't Fix</i>
SSB_5650526_329	CompoundX_Mainnet.sol 	L808 - L810	 <i>Won't Fix</i>
SSB_5650526_330	CompoundX_Mainnet.sol 	L813 - L821	 <i>Won't Fix</i>
SSB_5650526_331	CompoundX_Mainnet.sol 	L832 - L834	 <i>Won't Fix</i>
SSB_5650526_332	CompoundX_Mainnet.sol 	L837 - L844	 <i>Won't Fix</i>
SSB_5650526_333	CompoundX_Mainnet.sol 	L850 - L867	 <i>Won't Fix</i>
SSB_5650526_334	CompoundX_Mainnet.sol 	L854 - L859	 <i>Won't Fix</i>
SSB_5650526_335	CompoundX_Mainnet.sol 	L856 - L858	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_336	CompoundX_Mainnet.sol ↗	L859 - L866	 <i>Won't Fix</i>
SSB_5650526_337	CompoundX_Mainnet.sol ↗	L861 - L865	 <i>Won't Fix</i>
SSB_5650526_338	CompoundX_Mainnet.sol ↗	L862 - L864	 <i>Won't Fix</i>
SSB_5650526_339	CompoundX_Mainnet.sol ↗	L870 - L886	 <i>Won't Fix</i>
SSB_5650526_340	CompoundX_Mainnet.sol ↗	L871 - L874	 <i>Won't Fix</i>
SSB_5650526_341	CompoundX_Mainnet.sol ↗	L874 - L880	 <i>Won't Fix</i>
SSB_5650526_342	CompoundX_Mainnet.sol ↗	L875 - L877	 <i>Won't Fix</i>
SSB_5650526_343	CompoundX_Mainnet.sol ↗	L877 - L879	 <i>Won't Fix</i>
SSB_5650526_344	CompoundX_Mainnet.sol ↗	L894 - L901	 <i>Won't Fix</i>
SSB_5650526_345	CompoundX_Mainnet.sol ↗	L896 - L899	 <i>Won't Fix</i>
SSB_5650526_346	CompoundX_Mainnet.sol ↗	L903 - L906	 <i>Won't Fix</i>
SSB_5650526_347	CompoundX_Mainnet.sol ↗	L911 - L915	 <i>Won't Fix</i>
SSB_5650526_348	CompoundX_Mainnet.sol ↗	L912 - L914	 <i>Won't Fix</i>
SSB_5650526_349	CompoundX_Mainnet.sol ↗	L953 - L959	 <i>Won't Fix</i>
SSB_5650526_350	CompoundX_Mainnet.sol ↗	L973 - L1005	 <i>Won't Fix</i>
SSB_5650526_351	CompoundX_Mainnet.sol ↗	L980 - L982	 <i>Won't Fix</i>
SSB_5650526_351	CompoundX_Mainnet.sol ↗	L1026 - L1028	 <i>Won't Fix</i>
SSB_5650526_352	CompoundX_Mainnet.sol ↗	L994 - L1004	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_353	CompoundX_Mainnet.sol 	L997 - L999	 <i>Won't Fix</i>
SSB_5650526_354	CompoundX_Mainnet.sol 	L999 - L1001	 <i>Won't Fix</i>
SSB_5650526_355	CompoundX_Mainnet.sol 	L1019 - L1036	 <i>Won't Fix</i>
SSB_5650526_356	CompoundX_Mainnet.sol 	L1043 - L1051	 <i>Won't Fix</i>
SSB_5650526_357	CompoundX_Mainnet.sol 	L1048 - L1050	 <i>Won't Fix</i>

Issue Type



















MISSING NATSPEC DESCRIPTIONS FOR PUBLIC VARIABLE DECLARATIONS

S. No.	Severity	Detection Method	Instances
I014	● Informational	Automated	38

Description

Public variable declarations without NatSpec descriptions are found in the code. NatSpec comments improve readability and provide clear documentation for externally accessible variables. Without them, developers and users may struggle to understand the role and impact of these variables when interacting with the contract.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_132	CompoundX_Mainnet.sol ↗	L33 - L33	 <i>Won't Fix</i>
SSB_5650526_133	CompoundX_Mainnet.sol ↗	L35 - L35	 <i>Won't Fix</i>
SSB_5650526_134	CompoundX_Mainnet.sol ↗	L37 - L37	 <i>Won't Fix</i>
SSB_5650526_135	CompoundX_Mainnet.sol ↗	L38 - L38	 <i>Won't Fix</i>
SSB_5650526_136	CompoundX_Mainnet.sol ↗	L39 - L39	 <i>Won't Fix</i>
SSB_5650526_137	CompoundX_Mainnet.sol ↗	L40 - L40	 <i>Won't Fix</i>
SSB_5650526_138	CompoundX_Mainnet.sol ↗	L42 - L42	 <i>Won't Fix</i>
SSB_5650526_139	CompoundX_Mainnet.sol ↗	L43 - L43	 <i>Won't Fix</i>
SSB_5650526_140	CompoundX_Mainnet.sol ↗	L44 - L44	 <i>Won't Fix</i>
SSB_5650526_141	CompoundX_Mainnet.sol ↗	L45 - L45	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_142	CompoundX_Mainnet.sol ↗	L48 - L48	 <i>Won't Fix</i>
SSB_5650526_143	CompoundX_Mainnet.sol ↗	L49 - L49	 <i>Won't Fix</i>
SSB_5650526_144	CompoundX_Mainnet.sol ↗	L51 - L51	 <i>Won't Fix</i>
SSB_5650526_145	CompoundX_Mainnet.sol ↗	L52 - L52	 <i>Won't Fix</i>
SSB_5650526_146	CompoundX_Mainnet.sol ↗	L54 - L54	 <i>Won't Fix</i>
SSB_5650526_147	CompoundX_Mainnet.sol ↗	L55 - L55	 <i>Won't Fix</i>
SSB_5650526_148	CompoundX_Mainnet.sol ↗	L57 - L57	 <i>Won't Fix</i>
SSB_5650526_149	CompoundX_Mainnet.sol ↗	L59 - L59	 <i>Won't Fix</i>
SSB_5650526_150	CompoundX_Mainnet.sol ↗	L60 - L60	 <i>Won't Fix</i>
SSB_5650526_151	CompoundX_Mainnet.sol ↗	L61 - L61	 <i>Won't Fix</i>
SSB_5650526_152	CompoundX_Mainnet.sol ↗	L62 - L62	 <i>Won't Fix</i>
SSB_5650526_153	CompoundX_Mainnet.sol ↗	L63 - L63	 <i>Won't Fix</i>
SSB_5650526_154	CompoundX_Mainnet.sol ↗	L65 - L65	 <i>Won't Fix</i>
SSB_5650526_155	CompoundX_Mainnet.sol ↗	L66 - L66	 <i>Won't Fix</i>
SSB_5650526_156	CompoundX_Mainnet.sol ↗	L68 - L68	 <i>Won't Fix</i>
SSB_5650526_157	CompoundX_Mainnet.sol ↗	L69 - L69	 <i>Won't Fix</i>
SSB_5650526_158	CompoundX_Mainnet.sol ↗	L70 - L70	 <i>Won't Fix</i>
SSB_5650526_159	CompoundX_Mainnet.sol ↗	L115 - L115	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_160	CompoundX_Mainnet.sol 	L116 - L116	 <i>Won't Fix</i>
SSB_5650526_161	CompoundX_Mainnet.sol 	L117 - L117	 <i>Won't Fix</i>
SSB_5650526_162	CompoundX_Mainnet.sol 	L118 - L118	 <i>Won't Fix</i>
SSB_5650526_163	CompoundX_Mainnet.sol 	L119 - L119	 <i>Won't Fix</i>
SSB_5650526_164	CompoundX_Mainnet.sol 	L120 - L120	 <i>Won't Fix</i>
SSB_5650526_165	CompoundX_Mainnet.sol 	L122 - L122	 <i>Won't Fix</i>
SSB_5650526_166	CompoundX_Mainnet.sol 	L123 - L123	 <i>Won't Fix</i>
SSB_5650526_167	CompoundX_Mainnet.sol 	L124 - L124	 <i>Won't Fix</i>
SSB_5650526_168	CompoundX_Mainnet.sol 	L125 - L125	 <i>Won't Fix</i>
SSB_5650526_169	CompoundX_Mainnet.sol 	L127 - L127	 <i>Won't Fix</i>

Issue Type

MISSING @NOTICE IN NATSPEC COMMENTS FOR CONSTRUCTORS

S. No.	Severity	Detection Method	Instances
I015	● Informational	Automated	2



Description

Natspec `@notice` comments are missing from a constructor declaration in the code. The `@notice` tag provides an overview of what the constructor does, helping users and developers quickly understand its purpose.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_215	CompoundX_Mainnet.sol ↗	L12 - L14	<i>Won't Fix</i>
SSB_5650526_216	CompoundX_Mainnet.sol ↗	L141 - L153	<i>Won't Fix</i>









Issue Type



















MISSING @NOTICE IN NATSPEC COMMENTS FOR FUNCTIONS

















S. No.	Severity	Detection Method	Instances
I016	● Informational	Automated	42

Description

Natspec `@notice` comments are missing from a function declaration in the code. The `@notice` tag provides an overview of what the function does, helping users and developers quickly understand its purpose.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_9	CompoundX_Mainnet.sol ↗	L16 - L18	 <i>Won't Fix</i>
SSB_5650526_10	CompoundX_Mainnet.sol ↗	L25 - L28	 <i>Won't Fix</i>
SSB_5650526_11	CompoundX_Mainnet.sol ↗	L155 - L182	 <i>Won't Fix</i>
SSB_5650526_12	CompoundX_Mainnet.sol ↗	L184 - L234	 <i>Won't Fix</i>
SSB_5650526_13	CompoundX_Mainnet.sol ↗	L236 - L284	 <i>Won't Fix</i>
SSB_5650526_14	CompoundX_Mainnet.sol ↗	L286 - L343	 <i>Won't Fix</i>
SSB_5650526_15	CompoundX_Mainnet.sol ↗	L345 - L383	 <i>Won't Fix</i>
SSB_5650526_16	CompoundX_Mainnet.sol ↗	L385 - L431	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_17	CompoundX_Mainnet.sol ↗	L433 - L438	 <i>Won't Fix</i>
SSB_5650526_18	CompoundX_Mainnet.sol ↗	L440 - L445	 <i>Won't Fix</i>
SSB_5650526_19	CompoundX_Mainnet.sol ↗	L447 - L452	 <i>Won't Fix</i>
SSB_5650526_20	CompoundX_Mainnet.sol ↗	L454 - L459	 <i>Won't Fix</i>
SSB_5650526_21	CompoundX_Mainnet.sol ↗	L461 - L471	 <i>Won't Fix</i>
SSB_5650526_22	CompoundX_Mainnet.sol ↗	L473 - L483	 <i>Won't Fix</i>
SSB_5650526_23	CompoundX_Mainnet.sol ↗	L485 - L496	 <i>Won't Fix</i>
SSB_5650526_24	CompoundX_Mainnet.sol ↗	L498 - L521	 <i>Won't Fix</i>
SSB_5650526_25	CompoundX_Mainnet.sol ↗	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_26	CompoundX_Mainnet.sol ↗	L555 - L567	 <i>Won't Fix</i>
SSB_5650526_27	CompoundX_Mainnet.sol ↗	L569 - L593	 <i>Won't Fix</i>
SSB_5650526_28	CompoundX_Mainnet.sol ↗	L595 - L605	 <i>Won't Fix</i>
SSB_5650526_29	CompoundX_Mainnet.sol ↗	L607 - L624	 <i>Won't Fix</i>
SSB_5650526_30	CompoundX_Mainnet.sol ↗	L626 - L643	 <i>Won't Fix</i>
SSB_5650526_31	CompoundX_Mainnet.sol ↗	L645 - L647	 <i>Won't Fix</i>
SSB_5650526_32	CompoundX_Mainnet.sol ↗	L649 - L651	 <i>Won't Fix</i>
SSB_5650526_33	CompoundX_Mainnet.sol ↗	L653 - L669	 <i>Won't Fix</i>
SSB_5650526_34	CompoundX_Mainnet.sol ↗	L671 - L696	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_35	CompoundX_Mainnet.sol ↗	L698 - L701	 <i>Won't Fix</i>
SSB_5650526_36	CompoundX_Mainnet.sol ↗	L704 - L706	 <i>Won't Fix</i>
SSB_5650526_37	CompoundX_Mainnet.sol ↗	L708 - L717	 <i>Won't Fix</i>
SSB_5650526_38	CompoundX_Mainnet.sol ↗	L719 - L747	 <i>Won't Fix</i>
SSB_5650526_39	CompoundX_Mainnet.sol ↗	L749 - L776	 <i>Won't Fix</i>
SSB_5650526_40	CompoundX_Mainnet.sol ↗	L778 - L793	 <i>Won't Fix</i>
SSB_5650526_41	CompoundX_Mainnet.sol ↗	L795 - L827	 <i>Won't Fix</i>
SSB_5650526_42	CompoundX_Mainnet.sol ↗	L829 - L845	 <i>Won't Fix</i>
SSB_5650526_43	CompoundX_Mainnet.sol ↗	L847 - L887	 <i>Won't Fix</i>
SSB_5650526_44	CompoundX_Mainnet.sol ↗	L889 - L907	 <i>Won't Fix</i>
SSB_5650526_45	CompoundX_Mainnet.sol ↗	L909 - L917	 <i>Won't Fix</i>
SSB_5650526_46	CompoundX_Mainnet.sol ↗	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_47	CompoundX_Mainnet.sol ↗	L940 - L960	 <i>Won't Fix</i>
SSB_5650526_48	CompoundX_Mainnet.sol ↗	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_49	CompoundX_Mainnet.sol ↗	L1008 - L1037	 <i>Won't Fix</i>
SSB_5650526_50	CompoundX_Mainnet.sol ↗	L1039 - L1053	 <i>Won't Fix</i>

Issue Type

MISSING @NOTICE IN NATSPEC COMMENTS FOR MODIFIERS

S. No.	Severity	Detection Method	Instances
I017	● Informational	Automated	1



Description

Natspec `@notice` comments are missing from a modifier declaration in the code. The `@notice` tag provides a high-level description of a modifier's purpose, making it easier to understand its function. Lack of `@notice` documentation can reduce code readability, making it harder to understand restrictions applied by modifiers.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_381	CompoundX_Mainnet.sol ↗	L20 - L23	<i>Won't Fix</i>

Issue Type

MISSING UNDERSCORE IN NAMING VARIABLES

S. No.	Severity	Detection Method	Instances
I018	● Informational	Automated	1



Description

Solidity style guide suggests using underscores as the prefix for non-external functions and state variables (private or internal) but the contract was not found to be following the same.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_51	CompoundX_Mainnet.sol ↗	L473 - L483	Won't Fix













Issue Type

NAME MAPPING PARAMETERS

S. No.	Severity	Detection Method	Instances
I019	● Informational	Automated	6

Description

After Solidity 0.8.18, a feature was introduced to name mapping parameters. This helps in defining a purpose for each mapping and makes the code more descriptive.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_126	CompoundX_Mainnet.sol 	L115 - L115	 Won't Fix
SSB_5650526_127	CompoundX_Mainnet.sol 	L116 - L116	 Won't Fix
SSB_5650526_128	CompoundX_Mainnet.sol 	L117 - L117	 Won't Fix
SSB_5650526_129	CompoundX_Mainnet.sol 	L118 - L118	 Won't Fix
SSB_5650526_130	CompoundX_Mainnet.sol 	L119 - L119	 Won't Fix
SSB_5650526_131	CompoundX_Mainnet.sol 	L120 - L120	 Won't Fix


Issue Type

VARIABLES SHOULD BE IMMUTABLE

S. No.	Severity	Detection Method	Instances
I020	● Informational	Automated	1

Description

Constants and Immutables should be used in their appropriate contexts. `constant` should only be used for literal values written into the code. `immutable` variables should be used for expressions, or values calculated in, or passed into the constructor.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_358	CompoundX_Mainnet.sol ↗	L35 - L35	 <i>Won't Fix</i>

Issue Type

ARRAY LENGTH CACHING

S. No.	Severity	Detection Method	Instances
G001	● Gas	Automated	7

Description

During each iteration of the loop, reading the length of the array uses more gas than is necessary. In the most favorable scenario, in which the length is read from a memory variable, storing the array length in the stack can save about 3 gas per iteration. In the least favorable scenario, in which external calls are made during each iteration, the amount of gas wasted can be significant.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_359	CompoundX_Mainnet.sol 	L476 - L482	 <i>Won't Fix</i>
SSB_5650526_360	CompoundX_Mainnet.sol 	L559 - L564	 <i>Won't Fix</i>
SSB_5650526_361	CompoundX_Mainnet.sol 	L752 - L775	 <i>Won't Fix</i>
SSB_5650526_362	CompoundX_Mainnet.sol 	L801 - L826	 <i>Won't Fix</i>
SSB_5650526_363	CompoundX_Mainnet.sol 	L973 - L1005	 <i>Won't Fix</i>
SSB_5650526_364	CompoundX_Mainnet.sol 	L1019 - L1036	 <i>Won't Fix</i>
SSB_5650526_365	CompoundX_Mainnet.sol 	L1043 - L1051	 <i>Won't Fix</i>




Issue Type

ASSIGNING TO STRUCTS CAN BE MORE EFFICIENT

S. No.	Severity	Detection Method	Instances
G002	● Gas	Automated	3

Description

The contract is found to contain a struct with multiple variables defined in it. When a struct is assigned in a single operation, Solidity may perform costly storage operations, which can be inefficient. This often results in increased gas costs due to multiple SLOAD and SSTORE operations happening at once

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_414	CompoundX_Mainnet.sol ↗	L210 - L217	 Won't Fix
SSB_5650526_415	CompoundX_Mainnet.sol ↗	L326 - L333	 Won't Fix
SSB_5650526_416	CompoundX_Mainnet.sol ↗	L414 - L421	 Won't Fix


Issue Type

AVOID ZERO-TO-ONE STORAGE WRITES

S. No.	Severity	Detection Method	Instances
G003	● Gas	Automated	1

Description

Writing a storage variable from zero to a non-zero value costs 22,100 gas (20,000 for the write and 2,100 for cold access), making it one of the most expensive operations. This is why patterns like OpenZeppelin's `ReentrancyGuard` use `1` and `2` instead of `0` and `1`—to avoid the high cost of zero-to-non-zero writes. Non-zero to non-zero updates cost only 5,000 gas.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_429	CompoundX_Mainnet.sol ↗	L143 - L143	 <i>Won't Fix</i>


Issue Type



















CHEAPER CONDITIONAL OPERATORS

S. No.	Severity	Detection Method	Instances
G004	● Gas	Automated	17

Description

During compilation, `x != 0` is cheaper than `x > 0` for unsigned integers in solidity inside conditional statements.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_253	CompoundX_Mainnet.sol ↗	L156 - L156	 <i>Won't Fix</i>
SSB_5650526_254	CompoundX_Mainnet.sol ↗	L185 - L185	 <i>Won't Fix</i>
SSB_5650526_255	CompoundX_Mainnet.sol ↗	L237 - L237	 <i>Won't Fix</i>
SSB_5650526_255	CompoundX_Mainnet.sol ↗	L287 - L287	 <i>Won't Fix</i>
SSB_5650526_255	CompoundX_Mainnet.sol ↗	L386 - L386	 <i>Won't Fix</i>
SSB_5650526_256	CompoundX_Mainnet.sol ↗	L257 - L257	 <i>Won't Fix</i>
SSB_5650526_256	CompoundX_Mainnet.sol ↗	L308 - L308	 <i>Won't Fix</i>
SSB_5650526_256	CompoundX_Mainnet.sol ↗	L356 - L356	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_256	CompoundX_Mainnet.sol 	L397 - L397	 <i>Won't Fix</i>
SSB_5650526_257	CompoundX_Mainnet.sol 	L346 - L346	 <i>Won't Fix</i>
SSB_5650526_258	CompoundX_Mainnet.sol 	L941 - L941	 <i>Won't Fix</i>
SSB_5650526_259	CompoundX_Mainnet.sol 	L198 - L198	 <i>Won't Fix</i>
SSB_5650526_260	CompoundX_Mainnet.sol 	L267 - L267	 <i>Won't Fix</i>
SSB_5650526_261	CompoundX_Mainnet.sol 	L731 - L731	 <i>Won't Fix</i>
SSB_5650526_262	CompoundX_Mainnet.sol 	L789 - L789	 <i>Won't Fix</i>
SSB_5650526_263	CompoundX_Mainnet.sol 	L862 - L862	 <i>Won't Fix</i>
SSB_5650526_264	CompoundX_Mainnet.sol 	L953 - L953	 <i>Won't Fix</i>

Issue Type



































CHEAPER INEQUALITIES IN IF()

S. No.	Severity	Detection Method	Instances
G005	● Gas	Automated	25

Description

The contract was found to be doing comparisons using inequalities inside the if statement. When inside the `if` statements, non-strict inequalities (`>=`, `<=`) are usually cheaper than the strict equalities (`>`, `<`).

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_391	CompoundX_Mainnet.sol ↗	L198 - L198	 <i>Won't Fix</i>
SSB_5650526_392	CompoundX_Mainnet.sol ↗	L267 - L267	 <i>Won't Fix</i>
SSB_5650526_393	CompoundX_Mainnet.sol ↗	L448 - L448	 <i>Won't Fix</i>
SSB_5650526_394	CompoundX_Mainnet.sol ↗	L462 - L462	 <i>Won't Fix</i>
SSB_5650526_395	CompoundX_Mainnet.sol ↗	L467 - L467	 <i>Won't Fix</i>
SSB_5650526_396	CompoundX_Mainnet.sol ↗	L478 - L478	 <i>Won't Fix</i>
SSB_5650526_397	CompoundX_Mainnet.sol ↗	L561 - L561	 <i>Won't Fix</i>
SSB_5650526_398	CompoundX_Mainnet.sol ↗	L660 - L660	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_399	CompoundX_Mainnet.sol 	L731 - L731	 <i>Won't Fix</i>
SSB_5650526_400	CompoundX_Mainnet.sol 	L757 - L757	 <i>Won't Fix</i>
SSB_5650526_401	CompoundX_Mainnet.sol 	L759 - L759	 <i>Won't Fix</i>
SSB_5650526_402	CompoundX_Mainnet.sol 	L761 - L761	 <i>Won't Fix</i>
SSB_5650526_403	CompoundX_Mainnet.sol 	L789 - L789	 <i>Won't Fix</i>
SSB_5650526_404	CompoundX_Mainnet.sol 	L806 - L806	 <i>Won't Fix</i>
SSB_5650526_405	CompoundX_Mainnet.sol 	L813 - L813	 <i>Won't Fix</i>
SSB_5650526_406	CompoundX_Mainnet.sol 	L856 - L856	 <i>Won't Fix</i>
SSB_5650526_407	CompoundX_Mainnet.sol 	L860 - L860	 <i>Won't Fix</i>
SSB_5650526_408	CompoundX_Mainnet.sol 	L862 - L862	 <i>Won't Fix</i>
SSB_5650526_409	CompoundX_Mainnet.sol 	L920 - L920	 <i>Won't Fix</i>
SSB_5650526_410	CompoundX_Mainnet.sol 	L953 - L953	 <i>Won't Fix</i>
SSB_5650526_411	CompoundX_Mainnet.sol 	L963 - L963	 <i>Won't Fix</i>
SSB_5650526_411	CompoundX_Mainnet.sol 	L1009 - L1009	 <i>Won't Fix</i>
SSB_5650526_412	CompoundX_Mainnet.sol 	L980 - L980	 <i>Won't Fix</i>
SSB_5650526_412	CompoundX_Mainnet.sol 	L1026 - L1026	 <i>Won't Fix</i>
SSB_5650526_413	CompoundX_Mainnet.sol 	L1041 - L1041	 <i>Won't Fix</i>

Issue Type























CHEAPER INEQUALITIES IN REQUIRE()

S. No.	Severity	Detection Method	Instances
G006	● Gas	Automated	19

Description

The contract was found to be performing comparisons using inequalities inside the `require` statement. When inside the `require` statements, non-strict inequalities (`>=`, `<=`) are usually costlier than strict equalities (`>`, `<`).

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_366	CompoundX_Mainnet.sol ↗	L156 - L156	 <i>Won't Fix</i>
SSB_5650526_367	CompoundX_Mainnet.sol ↗	L168 - L168	 <i>Won't Fix</i>
SSB_5650526_368	CompoundX_Mainnet.sol ↗	L185 - L185	 <i>Won't Fix</i>
SSB_5650526_369	CompoundX_Mainnet.sol ↗	L186 - L186	 <i>Won't Fix</i>
SSB_5650526_370	CompoundX_Mainnet.sol ↗	L237 - L237	 <i>Won't Fix</i>
SSB_5650526_370	CompoundX_Mainnet.sol ↗	L287 - L287	 <i>Won't Fix</i>
SSB_5650526_370	CompoundX_Mainnet.sol ↗	L386 - L386	 <i>Won't Fix</i>
SSB_5650526_371	CompoundX_Mainnet.sol ↗	L263 - L263	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_371	CompoundX_Mainnet.sol 	L362 - L362	 <i>Won't Fix</i>
SSB_5650526_372	CompoundX_Mainnet.sol 	L264 - L264	 <i>Won't Fix</i>
SSB_5650526_372	CompoundX_Mainnet.sol 	L363 - L363	 <i>Won't Fix</i>
SSB_5650526_373	CompoundX_Mainnet.sol 	L314 - L314	 <i>Won't Fix</i>
SSB_5650526_374	CompoundX_Mainnet.sol 	L315 - L315	 <i>Won't Fix</i>
SSB_5650526_375	CompoundX_Mainnet.sol 	L346 - L346	 <i>Won't Fix</i>
SSB_5650526_376	CompoundX_Mainnet.sol 	L403 - L403	 <i>Won't Fix</i>
SSB_5650526_377	CompoundX_Mainnet.sol 	L404 - L404	 <i>Won't Fix</i>
SSB_5650526_378	CompoundX_Mainnet.sol 	L941 - L941	 <i>Won't Fix</i>
SSB_5650526_379	CompoundX_Mainnet.sol 	L943 - L943	 <i>Won't Fix</i>
SSB_5650526_380	CompoundX_Mainnet.sol 	L946 - L946	 <i>Won't Fix</i>

Issue Type









DEFAULT INT VALUES ARE MANUALLY RESET

S. No.	Severity	Detection Method	Instances
G007	● Gas	Automated	9

Description

The contract is found to inefficiently reset integer variables to their default value of zero using manual assignment. In Solidity, manually setting a variable to its default value does not free up storage space, leading to unnecessary gas consumption. Instead, using the `.delete` keyword can achieve the same result while also freeing up storage space on the Ethereum blockchain, resulting in gas cost savings.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_93	CompoundX_Mainnet.sol ↗	L369 - L369	 <i>Won't Fix</i>
SSB_5650526_93	CompoundX_Mainnet.sol ↗	L406 - L406	 <i>Won't Fix</i>
SSB_5650526_94	CompoundX_Mainnet.sol ↗	L370 - L370	 <i>Won't Fix</i>
SSB_5650526_94	CompoundX_Mainnet.sol ↗	L407 - L407	 <i>Won't Fix</i>
SSB_5650526_95	CompoundX_Mainnet.sol ↗	L371 - L371	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_95	CompoundX_Mainnet.sol 	L408 - L408	 <i>Won't Fix</i>
SSB_5650526_96	CompoundX_Mainnet.sol 	L878 - L878	 <i>Won't Fix</i>
SSB_5650526_97	CompoundX_Mainnet.sol 	L995 - L995	 <i>Won't Fix</i>
SSB_5650526_98	CompoundX_Mainnet.sol 	L1000 - L1000	 <i>Won't Fix</i>


Issue Type

DEFINE CONSTRUCTOR AS PAYABLE

S. No.	Severity	Detection Method	Instances
G008	● Gas	Automated	1

Description

Developers can save around 10 opcodes and some gas if the constructors are defined as payable. However, it should be noted that it comes with risks because payable constructors can accept ETH during deployment.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_8	CompoundX_Mainnet.sol ↗	L141 - L153	 Won't Fix



Issue Type

EMIT USED IN LOOP

S. No.	Severity	Detection Method	Instances
G009	● Gas	Automated	2

 **Description**

In Solidity, when a code emits an event inside of a loop, internally it performs a LOG operation N times, where N represents the number of iterations in the loop. This can lead to inflated gas costs and potentially impact the efficiency of the code.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_178	CompoundX_Mainnet.sol ↗	L767 - L772	 Won't Fix
SSB_5650526_179	CompoundX_Mainnet.sol ↗	L813 - L821	 Won't Fix

Issue Type

FUNCTIONS CAN BE IN-LINED

S. No.	Severity	Detection Method	Instances
G010	● Gas	Automated	7



Description

The internal function was called only once throughout the contract. Internal functions cost more gas due to additional `JUMP` instructions and stack operations.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_85	CompoundX_Mainnet.sol ↗	L473 - L483	<i>Won't Fix</i>
SSB_5650526_86	CompoundX_Mainnet.sol ↗	L485 - L496	<i>Won't Fix</i>
SSB_5650526_87	CompoundX_Mainnet.sol ↗	L653 - L669	<i>Won't Fix</i>
SSB_5650526_88	CompoundX_Mainnet.sol ↗	L778 - L793	<i>Won't Fix</i>
SSB_5650526_89	CompoundX_Mainnet.sol ↗	L795 - L827	<i>Won't Fix</i>
SSB_5650526_90	CompoundX_Mainnet.sol ↗	L829 - L845	<i>Won't Fix</i>
SSB_5650526_91	CompoundX_Mainnet.sol ↗	L1008 - L1037	<i>Won't Fix</i>













Issue Type

FUNCTION SHOULD RETURN STRUCT

S. No.	Severity	Detection Method	Instances
G011	● Gas	Automated	6

Description

The function was detected to be returning multiple values.
Consider using a `struct` instead of multiple return values for the function. It can improve code readability.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_209	CompoundX_Mainnet.sol 	L498 - L521	 <i>Won't Fix</i>
SSB_5650526_210	CompoundX_Mainnet.sol 	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_211	CompoundX_Mainnet.sol 	L607 - L624	 <i>Won't Fix</i>
SSB_5650526_212	CompoundX_Mainnet.sol 	L626 - L643	 <i>Won't Fix</i>
SSB_5650526_213	CompoundX_Mainnet.sol 	L671 - L696	 <i>Won't Fix</i>
SSB_5650526_214	CompoundX_Mainnet.sol 	L719 - L747	 <i>Won't Fix</i>






Issue Type



GAS INEFFICIENCY DUE TO MULTIPLE OPERANDS IN SINGLE IF/ELSEIF CONDITION

S. No.	Severity	Detection Method	Instances
G012	● Gas	Automated	9

Description

The contract is found to use multiple operands within a single `if` or `else if` statement, which can lead to unnecessary gas consumption due to the way the EVM evaluates compound boolean expressions. Each operand in a compound condition is evaluated even if the first condition fails, unless short-circuiting occurs, and the combined logic can result in more complex bytecode and higher gas usage compared to using nested `if` statements. This inefficiency is particularly relevant in functions that are called frequently or within loops.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_201	CompoundX_Mainnet.sol ↗	L196 - L203	 <i>Won't Fix</i>
SSB_5650526_202	CompoundX_Mainnet.sol ↗	L448 - L448	 <i>Won't Fix</i>
SSB_5650526_203	CompoundX_Mainnet.sol ↗	L462 - L462	 <i>Won't Fix</i>
SSB_5650526_204	CompoundX_Mainnet.sol ↗	L657 - L657	 <i>Won't Fix</i>
SSB_5650526_205	CompoundX_Mainnet.sol ↗	L731 - L736	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_206	CompoundX_Mainnet.sol 	L850 - L867	 <i>Won't Fix</i>
SSB_5650526_207	CompoundX_Mainnet.sol 	L963 - L963	 <i>Won't Fix</i>
SSB_5650526_207	CompoundX_Mainnet.sol 	L1009 - L1009	 <i>Won't Fix</i>
SSB_5650526_208	CompoundX_Mainnet.sol 	L1041 - L1041	 <i>Won't Fix</i>




Issue Type

GAS OPTIMIZATION FOR STATE VARIABLES

S. No.	Severity	Detection Method	Instances
G013	● Gas	Automated	3

Description

Plus equals (`+=`) costs more gas than addition operator. The same thing happens with minus equals (`-=`).

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_195	CompoundX_Mainnet.sol ↗	L221 - L221	 <i>Won't Fix</i>
SSB_5650526_196	CompoundX_Mainnet.sol ↗	L280 - L280	 <i>Won't Fix</i>
SSB_5650526_197	CompoundX_Mainnet.sol ↗	L377 - L377	 <i>Won't Fix</i>

Issue Type

GAS OPTIMIZATION IN DECREMENTS

S. No.	Severity	Detection Method	Instances
G014	● Gas	Automated	3



Description

`--i` costs less gas compared to `i--` or `i -= 1` for unsigned integers. In `i--`, the compiler has to create a temporary variable to store the initial value. This is not the case with `--i` in which the value is directly decremented and returned, thus, making it a cheaper alternative.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_251	CompoundX_Mainnet.sol ↗	L488 - L488	<i>Won't Fix</i>
SSB_5650526_251	CompoundX_Mainnet.sol ↗	L894 - L894	<i>Won't Fix</i>
SSB_5650526_252	CompoundX_Mainnet.sol ↗	L911 - L911	<i>Won't Fix</i>






Issue Type

GAS OPTIMIZATION IN INCREMENTS

S. No.	Severity	Detection Method	Instances
G015	● Gas	Automated	7

Description

`++i` costs less gas compared to `i++` or `i += 1` for unsigned integers. In `i++`, the compiler has to create a temporary variable to store the initial value. This is not the case with `++i` in which the value is directly incremented and returned, thus, making it a cheaper alternative.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_191	CompoundX_Mainnet.sol ↗	L476 - L476	 <i>Won't Fix</i>
SSB_5650526_191	CompoundX_Mainnet.sol ↗	L559 - L559	 <i>Won't Fix</i>
SSB_5650526_191	CompoundX_Mainnet.sol ↗	L973 - L973	 <i>Won't Fix</i>
SSB_5650526_191	CompoundX_Mainnet.sol ↗	L1019 - L1019	 <i>Won't Fix</i>
SSB_5650526_192	CompoundX_Mainnet.sol ↗	L752 - L752	 <i>Won't Fix</i>
SSB_5650526_193	CompoundX_Mainnet.sol ↗	L801 - L801	 <i>Won't Fix</i>
SSB_5650526_194	CompoundX_Mainnet.sol ↗	L1043 - L1043	 <i>Won't Fix</i>

Issue Type

LONG NUMBER LITERALS

S. No.	Severity	Detection Method	Instances
G016	● Gas	Automated	1



Description

Solidity supports multiple rational and integer literals, including decimal fractions and scientific notations. The use of very large numbers with too many digits was detected in the code that could have been optimized using a different notation also supported by Solidity.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_517	CompoundX_Mainnet.sol ↗	L70 - L70	Won't Fix

Issue Type

LONG REQUIRE/REVERT STRINGS

S. No.	Severity	Detection Method	Instances
G017	● Gas	Automated	1



Description

The `require()` and `revert()` functions take an input string to show errors if the validation fails. This strings inside these functions that are longer than `32 bytes` require at least one additional `MSTORE`, along with additional overhead for computing memory offset, and other parameters.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_125	CompoundX_Mainnet.sol ↗	L946 - L946	<i>Won't Fix</i>

Issue Type

NAMED RETURN OF LOCAL VARIABLE SAVES GAS AS COMPARED TO RETURN STATEMENT

S. No.	Severity	Detection Method	Instances
G018	● Gas	Automated	1



Description

The function having a return type is found to be declaring a local variable for returning, which causes extra gas consumption. This inefficiency arises because creating and manipulating local variables requires additional computational steps and memory allocation.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_474	CompoundX_Mainnet.sol ↗	L555 - L567	Won't Fix

Issue Type

PUBLIC CONSTANTS CAN BE PRIVATE

S. No.	Severity	Detection Method	Instances
G019	● Gas	Automated	20

Description

Public constant variables cost more gas because the EVM automatically creates getter functions for them and adds entries to the method ID table. The values can be read from the source code instead.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_57	CompoundX_Mainnet.sol ↗	L37 - L37	 <i>Won't Fix</i>
SSB_5650526_58	CompoundX_Mainnet.sol ↗	L38 - L38	 <i>Won't Fix</i>
SSB_5650526_59	CompoundX_Mainnet.sol ↗	L39 - L39	 <i>Won't Fix</i>
SSB_5650526_60	CompoundX_Mainnet.sol ↗	L40 - L40	 <i>Won't Fix</i>
SSB_5650526_61	CompoundX_Mainnet.sol ↗	L42 - L42	 <i>Won't Fix</i>
SSB_5650526_62	CompoundX_Mainnet.sol ↗	L43 - L43	 <i>Won't Fix</i>
SSB_5650526_63	CompoundX_Mainnet.sol ↗	L44 - L44	 <i>Won't Fix</i>
SSB_5650526_64	CompoundX_Mainnet.sol ↗	L45 - L45	 <i>Won't Fix</i>
SSB_5650526_65	CompoundX_Mainnet.sol ↗	L48 - L48	 <i>Won't Fix</i>
SSB_5650526_66	CompoundX_Mainnet.sol ↗	L49 - L49	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_67	CompoundX_Mainnet.sol ↗	L51 - L51	 <i>Won't Fix</i>
SSB_5650526_68	CompoundX_Mainnet.sol ↗	L52 - L52	 <i>Won't Fix</i>
SSB_5650526_69	CompoundX_Mainnet.sol ↗	L59 - L59	 <i>Won't Fix</i>
SSB_5650526_70	CompoundX_Mainnet.sol ↗	L60 - L60	 <i>Won't Fix</i>
SSB_5650526_71	CompoundX_Mainnet.sol ↗	L61 - L61	 <i>Won't Fix</i>
SSB_5650526_72	CompoundX_Mainnet.sol ↗	L62 - L62	 <i>Won't Fix</i>
SSB_5650526_73	CompoundX_Mainnet.sol ↗	L63 - L63	 <i>Won't Fix</i>
SSB_5650526_74	CompoundX_Mainnet.sol ↗	L65 - L65	 <i>Won't Fix</i>
SSB_5650526_75	CompoundX_Mainnet.sol ↗	L66 - L66	 <i>Won't Fix</i>
SSB_5650526_76	CompoundX_Mainnet.sol ↗	L127 - L127	 <i>Won't Fix</i>

Issue Type







SMALLER DATA TYPES COST MORE

S. No.	Severity	Detection Method	Instances
G020	● Gas	Automated	11

Description

Usage of smaller integer types such as `uint8`, `uint16`, `int8`, or `int16` in arithmetic operations incur additional gas costs compared to the default `uint` and `int` types, which are typically `uint256` and `int256` respectively.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_180	CompoundX_Mainnet.sol ↗	L489 - L490	 <i>Won't Fix</i>
SSB_5650526_181	CompoundX_Mainnet.sol ↗	L491 - L491	 <i>Won't Fix</i>
SSB_5650526_182	CompoundX_Mainnet.sol ↗	L752 - L752	 <i>Won't Fix</i>
SSB_5650526_183	CompoundX_Mainnet.sol ↗	L759 - L759	 <i>Won't Fix</i>
SSB_5650526_184	CompoundX_Mainnet.sol ↗	L761 - L761	 <i>Won't Fix</i>
SSB_5650526_185	CompoundX_Mainnet.sol ↗	L768 - L768	 <i>Won't Fix</i>
SSB_5650526_186	CompoundX_Mainnet.sol ↗	L771 - L771	 <i>Won't Fix</i>
SSB_5650526_187	CompoundX_Mainnet.sol ↗	L801 - L801	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_188	CompoundX_Mainnet.sol 	L837 - L837	 <i>Won't Fix</i>
SSB_5650526_189	CompoundX_Mainnet.sol 	L870 - L870	 <i>Won't Fix</i>
SSB_5650526_190	CompoundX_Mainnet.sol 	L895 - L896	 <i>Won't Fix</i>

Issue Type

SPLITTING REQUIRE STATEMENTS

S. No.	Severity	Detection Method	Instances
G021	● Gas	Automated	7

Description

Require statements when combined using operators in a single statement usually lead to a larger deployment gas cost but with each runtime calls, the whole thing ends up being cheaper by some gas units.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_173	CompoundX_Mainnet.sol ↗	L156 - L156	 Won't Fix
SSB_5650526_174	CompoundX_Mainnet.sol ↗	L185 - L185	 Won't Fix
SSB_5650526_175	CompoundX_Mainnet.sol ↗	L237 - L237	 Won't Fix
SSB_5650526_175	CompoundX_Mainnet.sol ↗	L287 - L287	 Won't Fix
SSB_5650526_175	CompoundX_Mainnet.sol ↗	L386 - L386	 Won't Fix
SSB_5650526_176	CompoundX_Mainnet.sol ↗	L346 - L346	 Won't Fix
SSB_5650526_177	CompoundX_Mainnet.sol ↗	L941 - L941	 Won't Fix

Issue Type

STORAGE VARIABLE CACHING IN MEMORY





































S. No.	Severity	Detection Method	Instances
G022	● Gas	Automated	42

































Description

The contract is using the state variable multiple times in the function.

SLOADs are expensive (100 gas after the 1st one) compared to MLOAD / MSTORE (3 gas each).

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_217	CompoundX_Mainnet.sol 	L155 - L182	 <i>Won't Fix</i>
SSB_5650526_217	CompoundX_Mainnet.sol 	L155 - L182	 <i>Won't Fix</i>
SSB_5650526_218	CompoundX_Mainnet.sol 	L184 - L234	 <i>Won't Fix</i>
SSB_5650526_218	CompoundX_Mainnet.sol 	L184 - L234	 <i>Won't Fix</i>
SSB_5650526_219	CompoundX_Mainnet.sol 	L236 - L284	 <i>Won't Fix</i>
SSB_5650526_220	CompoundX_Mainnet.sol 	L286 - L343	 <i>Won't Fix</i>
SSB_5650526_221	CompoundX_Mainnet.sol 	L345 - L383	 <i>Won't Fix</i>
SSB_5650526_221	CompoundX_Mainnet.sol 	L345 - L383	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_222	CompoundX_Mainnet.sol 	L385 - L431	 <i>Won't Fix</i>
SSB_5650526_223	CompoundX_Mainnet.sol 	L447 - L452	 <i>Won't Fix</i>
SSB_5650526_223	CompoundX_Mainnet.sol 	L447 - L452	 <i>Won't Fix</i>
SSB_5650526_224	CompoundX_Mainnet.sol 	L454 - L459	 <i>Won't Fix</i>
SSB_5650526_224	CompoundX_Mainnet.sol 	L454 - L459	 <i>Won't Fix</i>
SSB_5650526_225	CompoundX_Mainnet.sol 	L461 - L471	 <i>Won't Fix</i>
SSB_5650526_225	CompoundX_Mainnet.sol 	L461 - L471	 <i>Won't Fix</i>
SSB_5650526_226	CompoundX_Mainnet.sol 	L523 - L553	 <i>Won't Fix</i>
SSB_5650526_227	CompoundX_Mainnet.sol 	L569 - L593	 <i>Won't Fix</i>
SSB_5650526_228	CompoundX_Mainnet.sol 	L653 - L669	 <i>Won't Fix</i>
SSB_5650526_228	CompoundX_Mainnet.sol 	L653 - L669	 <i>Won't Fix</i>
SSB_5650526_229	CompoundX_Mainnet.sol 	L671 - L696	 <i>Won't Fix</i>
SSB_5650526_230	CompoundX_Mainnet.sol 	L704 - L706	 <i>Won't Fix</i>
SSB_5650526_231	CompoundX_Mainnet.sol 	L719 - L747	 <i>Won't Fix</i>
SSB_5650526_231	CompoundX_Mainnet.sol 	L719 - L747	 <i>Won't Fix</i>
SSB_5650526_232	CompoundX_Mainnet.sol 	L749 - L776	 <i>Won't Fix</i>
SSB_5650526_232	CompoundX_Mainnet.sol 	L749 - L776	 <i>Won't Fix</i>
SSB_5650526_233	CompoundX_Mainnet.sol 	L795 - L827	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_233	CompoundX_Mainnet.sol 	L795 - L827	 <i>Won't Fix</i>
SSB_5650526_234	CompoundX_Mainnet.sol 	L829 - L845	 <i>Won't Fix</i>
SSB_5650526_234	CompoundX_Mainnet.sol 	L829 - L845	 <i>Won't Fix</i>
SSB_5650526_235	CompoundX_Mainnet.sol 	L847 - L887	 <i>Won't Fix</i>
SSB_5650526_235	CompoundX_Mainnet.sol 	L847 - L887	 <i>Won't Fix</i>
SSB_5650526_236	CompoundX_Mainnet.sol 	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_236	CompoundX_Mainnet.sol 	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_236	CompoundX_Mainnet.sol 	L919 - L938	 <i>Won't Fix</i>
SSB_5650526_237	CompoundX_Mainnet.sol 	L940 - L960	 <i>Won't Fix</i>
SSB_5650526_237	CompoundX_Mainnet.sol 	L940 - L960	 <i>Won't Fix</i>
SSB_5650526_238	CompoundX_Mainnet.sol 	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_238	CompoundX_Mainnet.sol 	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_238	CompoundX_Mainnet.sol 	L962 - L1006	 <i>Won't Fix</i>
SSB_5650526_239	CompoundX_Mainnet.sol 	L1008 - L1037	 <i>Won't Fix</i>
SSB_5650526_239	CompoundX_Mainnet.sol 	L1008 - L1037	 <i>Won't Fix</i>
SSB_5650526_239	CompoundX_Mainnet.sol 	L1008 - L1037	 <i>Won't Fix</i>






Issue Type







SUPERFLUOUS EVENT FIELDS

S. No.	Severity	Detection Method	Instances
G023	● Gas	Automated	11

Description

`block.timestamp` and `block.number` are by default added to event information. Adding them manually costs extra gas.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_240	CompoundX_Mainnet.sol 	L181 - L181	 <i>Won't Fix</i>
SSB_5650526_241	CompoundX_Mainnet.sol 	L233 - L233	 <i>Won't Fix</i>
SSB_5650526_242	CompoundX_Mainnet.sol 	L283 - L283	 <i>Won't Fix</i>
SSB_5650526_243	CompoundX_Mainnet.sol 	L342 - L342	 <i>Won't Fix</i>
SSB_5650526_244	CompoundX_Mainnet.sol 	L382 - L382	 <i>Won't Fix</i>
SSB_5650526_245	CompoundX_Mainnet.sol 	L430 - L430	 <i>Won't Fix</i>
SSB_5650526_246	CompoundX_Mainnet.sol 	L771 - L771	 <i>Won't Fix</i>
SSB_5650526_247	CompoundX_Mainnet.sol 	L791 - L791	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_248	CompoundX_Mainnet.sol 	L820 - L820	 <i>Won't Fix</i>
SSB_5650526_249	CompoundX_Mainnet.sol 	L905 - L905	 <i>Won't Fix</i>
SSB_5650526_250	CompoundX_Mainnet.sol 	L958 - L958	 <i>Won't Fix</i>

Issue Type

UNNECESSARY CHECKED ARITHMETIC IN LOOP

S. No.	Severity	Detection Method	Instances
G024	● Gas	Automated	11

Description

Increments inside a loop could never overflow due to the fact that the transaction will run out of gas before the variable reaches its limits. Therefore, it makes no sense to have checked arithmetic in such a place.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_1	CompoundX_Mainnet.sol ↗	L476 - L476	 <i>Won't Fix</i>
SSB_5650526_1	CompoundX_Mainnet.sol ↗	L559 - L559	 <i>Won't Fix</i>
SSB_5650526_1	CompoundX_Mainnet.sol ↗	L973 - L973	 <i>Won't Fix</i>
SSB_5650526_1	CompoundX_Mainnet.sol ↗	L1019 - L1019	 <i>Won't Fix</i>
SSB_5650526_2	CompoundX_Mainnet.sol ↗	L480 - L480	 <i>Won't Fix</i>
SSB_5650526_3	CompoundX_Mainnet.sol ↗	L488 - L488	 <i>Won't Fix</i>
SSB_5650526_3	CompoundX_Mainnet.sol ↗	L894 - L894	 <i>Won't Fix</i>
SSB_5650526_4	CompoundX_Mainnet.sol ↗	L752 - L752	 <i>Won't Fix</i>
SSB_5650526_5	CompoundX_Mainnet.sol ↗	L801 - L801	 <i>Won't Fix</i>
SSB_5650526_6	CompoundX_Mainnet.sol ↗	L911 - L911	 <i>Won't Fix</i>

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_7	CompoundX_Mainnet.sol 	L1043 - L1043	 <i>Won't Fix</i>

Issue Type

UNNECESSARY DEFAULT VALUE INITIALIZATION

S. No.	Severity	Detection Method	Instances
G025	● Gas	Automated	1



Description

The contract was found to be initializing the value of the variable to its default value. This is redundant and not required.

Bug ID	File Location	Line No.	Action Taken
SSB_5650526_525	CompoundX_Mainnet.sol ↗	L55 - L55	<i>Won't Fix</i>

06. Scan History

● Critical ● High ● Medium ● Low ● Informational ● Gas

No	Date	Security Score	Scan Overview
----	------	----------------	---------------

1.	2026-05-11	98.56	● 0 ● 0 ● 0 ● 0 ● 293 ● 181
----	------------	--------------	-----------------------------

07. Disclaimer

The Reports neither endorse nor condemn any specific project or team, nor do they guarantee the security of any specific project. The contents of this report do not, and should not be interpreted as having any bearing on, the economics of tokens, token sales, or any other goods, services, or assets.

The security audit is not meant to replace functional testing done before a software release.

There is no warranty that all possible security issues of a particular smart contract(s) will be found by the tool, i.e., It is not guaranteed that there will not be any further findings based solely on the results of this evaluation.

Emerging technologies such as Smart Contracts and Solidity carry a high level of technical risk and uncertainty. There is no warranty or representation made by this report to any Third Party in regards to the quality of code, the business model or the proprietors of any such business model, or the legal compliance of any business.

In no way should a third party use these reports to make any decisions about buying or selling a token, product, service, or any other asset. It should be noted that this report is not investment advice, is not intended to be relied on as investment advice, and has no endorsement of this project or team. It does not serve as a guarantee as to the project's absolute security.

The assessment provided by SolidityScan 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. SolidityScan owes no duty to any third party by virtue of publishing these Reports.

As one audit-based assessment cannot be considered comprehensive, we always recommend proceeding with several independent manual audits including manual audit and a public bug bounty program to ensure the security of the smart contracts.