GUARDIAN

Synthetix

SNX Migration

Security Assessment

October 28th, 2024

Summary

Audit Firm Guardian

Prepared By Daniel Gelfand, Owen Thurm, Dogus Kose, Andreas Zottl, Zdravko Hristov, Nicholas Chew

Client Firm Synthetix

Final Report Date October 28, 2024

Audit Summary

Synthetix engaged Guardian to review the security of its synthetix. From the 5th of August to the 26th of August, a team of 6 auditors reviewed the source code in scope. All findings have been recorded in the following report.

For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

Issues Detected Throughout the engagement 7 High/Critical issues were uncovered and promptly remediated by the Synthetix team. Several issues impacted the fundamental behavior of the protocol, following their remediation Guardian believes the protocol to uphold the functionality described for the SNX Migration.

🔗 Blockchain network: **Ethereum**

✅ Verify the authenticity of this report on Guardian's GitHub: https://github.com/guardianaudits

📊 Code coverage & PoC test suite: https://github.com/GuardianAudits/synthetix-legacy-poc

Table of Contents

Project Overview

Project Summary

Project Name	Synthetix
Language	Solidity
Codebase	https://github.com/Synthetixio/synthetix-v3
Commit(s)	Initial: 2895dde7d3f2c112a6f60ee12686a380f8e6deb2 Final: 7e9157e8eda2e9f3df84a9c6b65d701484bf3faa

Audit Summary

Delivery Date	October 28, 2024
Audit Methodology	Static Analysis, Manual Review, Test Suite, Contract Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	7	0	0	1	0	6
● Medium	17	0	0	12	1	4
● Low	19	0	0	15	0	4

Audit Scope & Methodology

Vulnerability Classifications

Severity	Impact: *High*	Impact: *Medium*	Impact: *Low*
Likelihood: *High*	● Critical	● High	● Medium
Likelihood: *Medium*	● High	● Medium	● Low
Likelihood: *Low*	● Medium	● Low	● Low

Impact

High	Significant loss of assets in the protocol, significant harm to a group of users, or a core functionality of the protocol is disrupted.
Medium	A small amount of funds can be lost or ancillary functionality of the protocol is affected. The user or protocol may experience reduced or delayed receipt of intended funds.
Low	Can lead to any unexpected behavior with some of the protocol's functionalities that is notable but does not meet the criteria for a higher severity.

Likelihood

High	The attack is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount gained or the disruption to the protocol.
Medium	An attack vector that is only possible in uncommon cases or requires a large amount of capital to exercise relative to the amount gained or the disruption to the protocol.
Low	Unlikely to ever occur in production.

Audit Scope & Methodology

Methodology

Guardian is the ultimate standard for Smart Contract security. An engagement with Guardian entails the following:

- Two competing teams of Guardian security researchers performing an independent review.
- A dedicated fuzzing engineer to construct a comprehensive stateful fuzzing suite for the project.
- An engagement lead security researcher coordinating the 2 teams, performing their own analysis, relaying findings to the client, and orchestrating the testing/verification efforts.

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. Comprehensive written tests as a part of a code coverage testing suite.
- Contract fuzzing for increased attack resilience.

Findings & Resolutions

ID	Title	Category	Severity	Status
H-01	Incorrect Debt Update During Migration	Logical Error	● High	Resolved
H-02	convertUSD DOS Because Of Share Rounding	DoS	● High	Resolved
H-03	Liquidation Of Contract Accounts May Be Blocked	Logical Error	● High	Resolved
H-04	Debt Burden After Vault Liquidation	Logical Error	● High	Acknowledged
H-05	Migrations Can Be Executed Even If Paused	Unexpected Behavior	● High	Resolved
H-06	Prevention Of convertUSD Via Bridge Migrations	DoS	● High	Resolved
H-07	Partial Vault Liquidations Ignore Rebalance	Logical Error	● High	Resolved
M-01	reportedDebt Could Report Stale Debt	Logical Error	● Medium	Acknowledged
M-02	Liquidations Prevented By Frontrunning	Logical Error	● Medium	Acknowledged
M-03	Collateral Locks Are Not Cleared On Liquidation	Unexpected Behavior	● Medium	Resolved
M-04	Missing Check In Migrate Flow	Validation	● Medium	Resolved
M-05	Missing Check In delegateCollateral	Validation	● Medium	Resolved
M-06	Claim FeePool Rewards During Migration	Logical Error	● Medium	Acknowledged

Findings & Resolutions

ID	Title	Category	Severity	Status
M-07	Associate Debt Incompatible With Multiple Pools	Unexpected Behavior	● Medium	Resolved
M-08	Some Liquidations Will Accrue Bad Debt	Logical Error	● Medium	Acknowledged
M-09	Incentives To Migrate Liquidatable Stakers	Logical Error	● Medium	Acknowledged
M-10	Migrations May Lock Other Markets	Unexpected Behavior	● Medium	Acknowledged
M-11	sUSD Wrapper Allows Burning Of All DebtShares	Logical Error	● Medium	Acknowledged
M-12	Legacy Market Unable To Claim SNX Rewards	Logical Error	● Medium	Acknowledged
M-13	Delegations Should Not Toggle Pools	Unexpected Behavior	● Medium	Acknowledged
M-14	burnSynthsToTarget Functionality Changes	Unexpected Behavior	● Medium	Acknowledged
M-15	Users Lose Their Rewards Upon Liquidation	Logical Error	● Medium	Partially Resolved
M-16	LegacyMarket Does Not Lock Collateral	Logical Error	● Medium	Acknowledged
M-17	Accounts Without Debt Stuck In V2	Logical Error	● Medium	Acknowledged
L-01	FeePool Fees Can Be Claimed With Bad C-ratio	Unexpected Behavior	● Low	Acknowledged
L-02	Inflated Rewards For LegacyMarket	Unexpected Behavior	● Low	Acknowledged

Findings & Resolutions

ID	Title	Category	Severity	Status
L-03	1 Week Lock Of Free Collaterals After Migration	DoS	● Low	Acknowledged
L-04	_calculateDebtValueMigrated May Divide By 0	Unexpected Behavior	● Low	Resolved
L-05	Excessive Escrows Instantly Claimable	Unexpected Behavior	● Low	Acknowledged
L-06	Migrated Funds May Not Be Withdrawable	Documentation	● Low	Resolved
L-07	TrustedMulticallForwarder Incompatibility	Unexpected Behavior	● Low	Acknowledged
L-08	Forced Exposure To Back BFP And Spot Market	Unexpected Behavior	● Low	Acknowledged
L-09	distributeDebtToPools Calls Not Future Proof	DoS	● Low	Acknowledged
L-10	_calculateDebtValueMigrated Should Round Up	Logical Error	● Low	Acknowledged
L-11	Market Weights And Monitoring	Warning	● Low	Acknowledged
L-12	Missing Check In mintUsd	Validation	● Low	Resolved
L-13	Min Delegation Amount Can Prevent Liquidations	Logical Error	● Low	Acknowledged
L-14	Stablecoin Supply Can Be Inflated	Logical Error	● Low	Resolved
L-15	reportedDebt Function Does Not Revert	Validation	● Low	Acknowledged

Findings & Resolutions

ID	Title	Category	Severity	Status
L-16	Setting issuanceRatio To 1 Locks Every Token	Documentation	● Low	Acknowledged
L-17	minDelegation May Prevent Migration	Unexpected Behavior	● Low	Acknowledged
L-18	Liquidator DoS Griefing Attack	DoS	● Low	Acknowledged
L-19	Market creditCapacity Mis-accounting	Documentation	● Low	Acknowledged

H-01 | Incorrect Debt Update During Migration

Category	Severity	Location	Status
Logical Error	● High	Global	Resolved

Description PoC

When the preferred Spartan Council pool contains multiple supported collateral type vaults the migration of a V2 staker's debt incorrectly deducts all of their associated debt from the collateral vault of the staker.

However other collateral vaults may burden a portion of the `reportedDebt` delta which was experienced by the `LegacyMarket` when the staker migrated.

For example (contrived example for simplicity):
• User A migrates their account which has X debt
• Upon a normal debt distribution X/2 of this debt would go to the `USDe` vault and X/2 to the SNX vault
• The Spartan Pool has two vaults: USDe and SNX, split 50%/50% in terms of backing liquidity for the Legacy Market
• The `AssociateDebt` module however deducts X debt from only the SNX vault with `distributeDebtToAccounts`
• As a result the `USDe` vault delegators experience a loss due to the burden of X/2 debt which should have been corrected for.

Recommendation

When associating debt for a pool with multiple backing vaults, reduce the amount of debt from all vaults corresponding to their contribution to the market. This way the `associatedDebt` is correctly drawn from all counterparties and correctly associated with the target account.

Resolution

Synthetix Team: The issue was resolved in commit d3a9e6d.

H-02 | convertUSD DOS Because Of Share Rounding

Category	Severity	Location	Status
DoS	● High	LegacyMarket	Resolved

Description PoC

LegacyMarket.convertUSD() burns synths from the converter's account and enforces a maximum difference between the actual burnt amount and the amount specified by the user to be no more than 1 unit of sUSD.

However, when the Issuer does the burning of the SDS, there is some rounding which happens on the shares. This means the end result can be with 1 unit less of a share, not sUSD.

Furthermore, additional rounding can happen on this line when querying the SDS balance which will increase the margin of error.

As a result, the burnt amount may deviate slightly more than 1 wei of sUSD when the `debtShares` ratio is above 1$ of debt per share. As the `LegacyMarket` does not tolerate this amount of inaccuracy, the whole transaction will revert and converting will not be possible.

Recommendation

Consider allowing for a rounding error of 1 wei of shares rather than 1 wei of USD.

Resolution

Synthetix Team: The issue was resolved in commit 5701e75.

H-03 | Liquidation Of Contract Accounts May Be Blocked

Category	Severity	Location	Status
Logical Error	● High	LegacyMarket.sol: 281	Resolved

Description

If an account is liquidatable on V2X, it is expected to be forcibly migrated to the V3 system where it will be liquidated. It cannot be liquidated on V2X.

However, if the account is a contract that does not implement `onERC721Received`, then the V3 account cannot be transferred and the migration would fail. This allows contract accounts to avoid liquidation and create bad debt within the V2X system.

Recommendation

Consider allowing the migration to proceed without an account transfer such that liquidation can continue, then provide some sort of claim mechanism for the account owner to recover remaining funds.

Resolution

Synthetix Team: The issue was resolved in commit 3617715.

H-04 | Debt Burden After Vault Liquidation

Category	Severity	Location	Status
Logical Error	● High	Global	Acknowledged

Description PoC

Vault liquidations are quite profitable for the liquidator who pays back the V3 debt but receives all collateral in the vault.

As `migrateOnBehalf` can be called by anyone, an attacker may be able to put the SNX vault in a liquidatable state by migrating the right positions from V2, which would result in the c-ratio of the vault dropping below the `collateralConfig.liquidationRatioD18` and become open for vault liquidation.

This would be especially possible if only a handful of small accounts were migrated to V3, and then a big liquidatable account is force migrated to make the entire vault liquidatable. In the event of a vault liquidation, all debts are settled and the liquidator takes all collateral from the vault.

Assuming there is only one vault in the pool, then there would be no collateral backing the debt shares that the Legacy market holds. This state in itself is somewhat peculiar and should be examined further.

One concern would be during the period after vault liquidation till the next time another user migrate. If debt shares increase in value during that period, then the next user to migrate take on all extra debt.

In many cases, any subsequent migrators/delegators would become immediately liquidatable by unfairly taking on debt that did not belong to them.

Recommendation

Ensure sufficient volume of healthy accounts are migrated first to ensure that vault liquidation is very unlikely to happen. Also, consider handling the scenario after vault liquidation where the pool goes out of range for a market, and do not attribute any increase in debt to the next user that migrates.

Resolution

Synthetix Team: Mainnet already has enough liquidity to absorb initial migrations.

H-05 | Migrations Can Be Executed Even If Paused

Category	Severity	Location	Status
Unexpected Behavior	● High	LegacyMarket.sol	Resolved

Description

LegacyMarket.migrate() reverts if pauseMigration = true . However, there is another function for migrations which lacks this check - migrateOnBehalf(). Anyone can call migrateOnBehalf and pass their address to execute a migration even when they are paused.

Recommendation

Move the check for paused migrations in the internal _migrate() function

Resolution

Synthetix Team: The issue was resolved in commit 6220373.

H-06 | Prevention Of convertUSD Via Bridge Migrations

Category	Severity	Location	Status
DoS	● High	LegacyMarket.sol: 161	Resolved

Description

The convertUSD function calls burnSynths in the V2 system which will revert if the minimum stake time has not passed since the last issuance event of the staker.

By bridging sUSD from an L2 to the L1 chain with the legacy market as the destination address, anyone can set the last issuance event of the legacy market to the current block time and therefore DoS the convertUSD function for one week.

This call can be repeated once per week which can make DOS permanent. Notice also that this same attack vector could be used to prevent users from burning to make their positions healthy.

Recommendation

Consider setting the SETTING_MINIMUM_STAKE_TIME to zero as no more synths can be freshly issued. Otherwise specifically disallow bridges to the LegacyMarket.

Resolution

Synthetix Team: Resolved.

H-07 | Partial Vault Liquidations Ignore Rebalance

Category	Severity	Location	Status
Logical Error	● High	LiquidationModule.sol: 185	Resolved

Description PoC

When liquidating a vault partially the markets are not rebalanced after the collateralLiquidated is seized by the liquidator.

Therefore markets are not aware of this reduction in creditCapacity and cannot accurately determine if they are below the minimumCredit threshold, which may be likely if a vault has been partially liquidated.

For example, BFP market and Perps Market rely on minimumCredit validations to decide whether increase orders should be allowed.

Additionally, the vaultsDebtDistribution shares are not updated for the vault after the collateral value changes, which can lead to further mis-accounting of credit and debt spread across vaults.

Recommendation

Be sure to call recalculateVaultCollateral function after the collateral is deducted from accounts in the partial vault liquidation case.

Resolution

Synthetix Team: The issue was resolved in commit 3a324b5.

M-01 | reportedDebt Could Report Stale Debt

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol: 101	Acknowledged

Description

The reportedDebt function obtains V2X system debt through debtBalanceOf which does not return the boolean for stale oracle rates.

If the rates are stale (e.g. if oracle was frozen), then an inaccurate debt value will be reported to the V3 system which V3 stakers could take advantage of.

For example, if the bulk of synths are in ETH, and the price of ETH doubles while the rate is stale, V3 stakers could undelegate in anticipation of a large debt increase to avoid socialization of debt among them.

Recommendation

Async delegation should address any risk of arbitrage with outdated pricing. Be aware of this risk and closely monitor any oracle outages to take necessary precautions to limit arbitrageable value as a result of stale prices.

Resolution

Synthetix Team: Acknowledged.

M-02 | Liquidations Prevented By Frontrunning

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol: 237	Acknowledged

Description

If an account is liquidatable on V2X, it is expected to be forcibly migrated to the V3 system where it will be liquidated. It cannot be liquidated on V2X.

However, `migrateOnBehalf` can be DOS-ed by front-running the liquidator and creating an account in the V3 system with the requested accountId. Then the `migrateOnBehalf` call would fail and revert since an identical `accountId` already exists.

Apart from gas costs, there is no limitation to how many times an account can be created to perform this DOS attack and avoid liquidation, which could result in bad debt to the system.

Recommendation

Re-consider allowing users to pass in `requestedAccountId` in `createAccount` in the V3 system and instead use the `createAccount()` function which automatically assigns the next free account number. Otherwise be sure to document that liquidations should use MEV protection such as flash-bots.

Resolution

Synthetix Team: Acknowledged.

M-03 | Collateral Locks Are Not Cleared On Liquidation

Category	Severity	Location	Status
Unexpected Behavior	● Medium	Global	Resolved

Description

When a migration happens, all of the user's collateral in escrows in v2 is locked in the v3 system, which means that it can be delegated, but not withdrawn.

These locks are not cleared on account liquidation. So if a user gets liquidated and deposits new collateral back into the same account afterwards, the unexpired locks will still prevent them from withdrawing this new collateral.

The same applies to vault liquidations. This may be unexpected for users and result in unexpected locked funds as a result of their previous locks in the V2 system, which should not apply to new collateral they deposit directly into V3.

Recommendation

When an account liquidation happens, iterate over the account's locks and deduct the liquidated amount from them. However, this will not work for vault liquidations because there will be many users in a vault. Either implement a different lock mechanism or document this risk.

Resolution

Synthetix Team: The issue was resolved in commit 82b58a3.

M-04 | Missing Check In Migrate Flow

Category	Severity	Location	Status
Validation	● Medium	LegacyMarket.sol: 196-288	Resolved

Description

• Liquidating a vault can be very profitable for the liquidator.
• The _migrate function does not check if the vault is liquidatable.

This allows to bring the vault into an unhealthy state on purpose, or to increase the profit for the liquidator further:

• The migration of an unhealthy V2 position can bring the SNX vault into an unhealthy state.
• It is possible to migrate more assets into a vault that is already liquidatable to increase the profit further.

Recommendation

Revert at the end of the migration flow if the vault is liquidatable.

Resolution

Synthetix Team: The issue was resolved in commit 78c0e2c.

M-05 | Missing Check In delegateCollateral

Category	Severity	Location	Status
Validation	● Medium	VaultModule.sol: 43-163	Resolved

Description

The `delegateCollateral` function does not check if the given vault can be liquidated. Therefore it is possible that an LP delegates to a vault and is instantly liquidated. This is a general issue in the `delegateCollateral` function, that can also occur when migrating from V2 to V3.

Recommendation

Add a check in the `delegateCollateral` function to ensure that the vault can not be liquidated.

Resolution

Synthetix Team: The issue was resolved in commit c88847d.

M-06 | Claim FeePool Rewards During Migration

Category	Severity	Location	Status
Logical Error	● Medium	Global	Acknowledged

Description

Unclaimed SNX rewards in V2X `FeePool` are rolled over each week and can be claimed by any debt shareholder. Some of the larger debt shareholders can claim an estimated 300 SNX each week, until all rolled over rewards are exhausted.

During migration, liquidator rewards are claimed for the migrating account but these `FeePool` rewards are not. If the accounts are force migrated, they lose the opportunity to claim before migrating and cannot claim after as they would no longer hold debt shares.

These rewards should be claimed as part of the migrating process to ensure all SNX owed to the account are moved to the V3 system.

Recommendation

Be sure to inform users about how the migration will affect the `FeePool` rewards, and that they should either claim all fee pool rewards before the `issuanceRatio` is update to 1 wei, or after migrating.

Resolution

Synthetix Team: Acknowledged.

M-07 | Associate Debt Incompatible With Multiple Pools

Category	Severity	Location	Status
Unexpected Behavior	● Medium	Legacy Market	Resolved

Description

The `AssociateDebtModule` allows a market to associate debt with a specific position. Based on the current implementation with the Legacy Market, it makes the assumption that all debt will be distributed to one pool.

However, if there are multiple pools backing the market, then the debt is proportionally distributed to the pools and `AssociateDebtModule` would incorrectly associate all the debt to the user in one pool.

Recommendation

If a market is connected to multiple pools, 1) associate the correct proportion of debt to the user in each pool/vault that the user is in, 2) perform a debt correction for pools which have received some of the debt.

Resolution

Synthetix Team: The issue was resolved in commit d3a9e6d.

M-08 | Some Liquidations Will Accrue Bad Debt

Category	Severity	Location	Status
Logical Error	● Medium	Global	Acknowledged

Description

The liquidation ratio for SNX token is specified as 1.05e18 which means liquidations will only be possible if an account is holding less than 1.05X collateral against 1.00X debt.

`liquidationRewardD18` is specified as 50 SNX token. When liquidating, what will be distributed to other LP's is: `collateralLiquidated - liquidationReward`. If this amount ever becomes less than `debtLiquidated`, then it will accrue bad debt to other LP's.

Considering the current configurations, this means any account liquidated that has less than 1050 SNX as collateral will cause other LP's to accrue bad debt.

Recommendation

Either increase the liquidation ratio so that bad debt accruing will be less likely to achieved (doesn't solve the problem completely), or introduce a dynamic liquidation reward mechanism that will both incentivize the liquidator but don't create bad debt.

Resolution

Synthetix Team: Simply raise the limit to ensure the 105% liquidation ratio doesn’t become a problem.

M-09 | Incentives To Migrate Liquidatable Stakers

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol: 189	Acknowledged

Description

While migration uses a significant amount of gas, there is no incentive to migrate liquidatable users. Only incentive is liquidation reward (50 tokens currently) which won't be sufficient especially when gas fees are high, hence liquidatable accounts won't be migrated.

Recommendation

Reward the migrator if the migration is done for liquidatable account in the `migrate()` by at least gas fee they have to pay.

Resolution

Synthetix Team: Acknowledged.

M-10 | Migrations May Lock Other Markets

Category	Severity	Location	Status
Unexpected Behavior	● Medium	Global	Acknowledged

Description PoC

In the V3 system all markets which are connected to a single pool affect each other's creditCapacity with the amount of debt they report.

This is because the totalVaultDebtsD18 is distributed evenly amongst all markets with the debtPerShareD18 and effectiveMaxShareValueD18 calculations in the rebalanceMarketsInPool function.

This is the expected architecture of the V3 system, however when performing migrations this allows a large migration of debt to potentially lock other markets as their minimumCredit threshold can be broken.

Normally healthy accounts will increase the delegated creditCapacity to other markets. However due to the minLiquidityRatio the creditCapacity that is allowed to be delegated to markets may actually be reduced after a migration.

In the worst case the other markets will become locked as their creditCapacity goes below the minimumCredit.

Recommendation

On the engagement kickoff it was mentioned that the BFP market and Spot market would be attached to the spartan council pool during the migration, however consider keeping the Legacy Market attached to a pool all by itself to avoid locking other markets until the migration is completed.

Otherwise consider validating that the migration does not put the pool in a capacity locked state, e.g. with the pool.findMarketWithCapacityLocked() function.

Resolution

Synthetix Team: Acknowledged.

M-11 | sUSD Wrapper Allows Burning Of All DebtShares

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol	Acknowledged

Description

sUSD can be minted from wrappers by depositing collateral such as USDe and receiving sUSD. An attacker could use these wrappers and the `convertUSD` function to burn all `debtShares` held by legacy market: `sUSD convert -> snxUSD, snxUSD exchange swap -> USDe, USDe wrap -> sUSD, repeat`.

This would be problematic if most or all debt shares have been migrated, and were burnt through exploit described above.

Then the leftover synths in the V2X system would be backed only by wrappers which traders cannot earn profit against (traders typically benefit from debt share value increase).

Recommendation

Consider restricting `convertUSD` only to accounts which have migrated, and limiting conversion up to their migrated debt amount.

Resolution

Synthetix Team: Acknowledged.

M-12 | Legacy Market Unable To Claim SNX Rewards

Category	Severity	Location	Status
Logical Error	● Medium	Global	Acknowledged

Description

Unclaimed SNX rewards in V2X FeePool are rolled over each week and can be claimed by any debt shareholder. As a large debt shareholder, Legacy Market can claim an estimated 500 SNX tokens each week, until all rolled over rewards are exhausted (around 2000 tokens remain).

However, Legacy Market currently lacks the ability (no implemented functions) to claim and vest these tokens from V2X FeePool. In the worst case, if all debt shares are migrated then these tokens will permanently remain unclaimed in the FeePool.

Recommendation

Consider adding functions to allow Legacy Market to claim and vest of these rewards.

Resolution

Synthetix Team: Acknowledged.

M-13 | Delegations Should Not Toggle Pools

Category	Severity	Location	Status
Unexpected Behavior	● Medium	Global	Acknowledged

Description PoC

A pool is disconnected from a market and moved to 'outRange' after it has hit its maxValuePerShare usually through changes in market performance.

However, it was observed that delegations may also move a pool in or out of range. In the case of a delegation (deposit), it was observed that the new delegator takes on the portion of the debt which was previous capped due to the pool being out of range.

This appears to be unexpected for a delegator to immediately gain debt from the pool after delegating and in the worst case, the delegator could become instantly liquidatable from the debt.

Recommendation

Consider re-examining the inRange/outRange mechanism to ensure that it can only move markets in or out based on market performance and not delegation as measured by creditCapacity.

Resolution

Synthetix Team: Acknowledged.

M-14 | burnSynthsToTarget Functionality Changes

Category	Severity	Location	Status
Unexpected Behavior	● Medium	Issuer.sol:L717	Acknowledged

Description

According to [SCCP-2134: LegacyMarket V3 Migration Preparations](), `issuanceRatio` will be settled to 1 wei. When `issuanceRatio` became 1 wei, `maxIssuableSynths` will be in the orders of wei for every user.

This will lead to `burnSynthsToTarget` function to leave its' normal functionality (which is burning to the c-ratio), and instead burn all debt user has.

Considering self-liquidations are disabled and right now it was announced to users that they need to burn back to c-ratio in order to avoid an automated liquidation, a lot of users might use `burnSynthsToTarget` function to burn to the c-ratio.

But with `issuanceRatio` change, they will encounter a result that they weren't expecting.

Recommendation

Inform users about this following change in `burnSynthsToTarget` so that they encounter not expected burns.

Resolution

Synthetix Team: Acknowledged.

M-15 | Users Lose Their Rewards Upon Liquidation

Category	Severity	Location	Status
Logical Error	● Medium	LiquidationModule.sol	Partially Resolved

Description

During the liquidation of an account, or a vault liquidation users lose their current vested, but unclaimed, RewardDistributor rewards as these amounts are based upon the shares of the current vault epoch.

Additionally, as the claimRewards function relies on the vault.currentEpoch() to measure the shares, reward amounts may be perturbed as the vault is liquidated and completely new shares are issued.

Recommendation

Consider claiming a user's rewards for their account on an individual liquidation. For vault liquidations, consider using the epoch in which the reward was applicable to measure the rewards earned, this way accounts can still claim their rewards which were attributed for that previous epoch.

Resolution

Synthetix Team: The issue was resolved in commit 6343ac5.

M-16 | LegacyMarket Does Not Lock Collateral

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol: 128	Acknowledged

Description

The LegacyMarket implementation of the minimumCredit function is hardcoded to return 0, therefore collateral is never locked on behalf of the LegacyMarket.

However the creditCapacity of the LegacyMarket is crucial for the migration of V2 sUsd to V3 snxUsd. The convertUsd function uses the withdrawMarketUsd function which reduces the market's creditCapacity by the withdrawn amount.

If the Legacy Market's creditCapacity is less than the value of debt which has been migrated, then the full debt cannot be migrated to the V3 system.

In the most explicative case, a single staker migrates $1,000 of SNX and $200 of debt and undelegates their collateral as there is no minimumCredit requirement. The immediate debt has been correctly accounted for and the staker is only able to withdraw a net value of $800.

However there remain 2 concerns:
1. The debtShares are now unbacked and further fluctuations in the debt value will not be covered by any collateral
2. The entire existing debt value cannot be migrated with the convertUsd function, as the creditCapacity has been reduced to $200.

Both of these issues raise concerns over the backing value and therefore the price peg for synths in the V2 system. Additionally, the second issue directly prevents the migration from being fully carried out.

Recommendation

Consider setting the minimumCredit to a nonzero value to ensure that sufficient collateral is present in the V3 pool to safely carry out the migration. Currently the creditCapacity of a market does not reduce as new totalDebt is added as mentioned in H-10.

With this behavior it may make sense to assign the minimumCredit to the reportedDebt as this is the value of debt which must be convertible with the convertUsd function. However, this current behavior is possibly flawed, in which case another solution for the minimumCredit will have to be constructed, depending on the resolution of H-10.

Resolution

Synthetix Team: Acknowledged.

M-17 | Accounts Without Debt Stuck In V2

Category	Severity	Location	Status
Logical Error	● Medium	LegacyMarket.sol: 308-310	Acknowledged

Description

It is possible for an account to have no debt but both liquid collateral and escrowed entries. This can occur with multiple ways, for example they might paid their debt by burning.

With a quick scan we were able to identify good amount of accounts that are in this position, which shows that we can assume there are much more.

These accounts are not migratable because of the check in `_gatherFromV2` which reverts with `NothingToMigrateerror`. These accounts can not create new debt as well because issuing is stopped.

Hence these accounts have to wait for their locks to expire in order to be able to utilize them, until then their tokens will be locked with no utilization.

Recommendation

Two possible ways to resolve this issue:
1. Create a new function to only migrate locks from V2 to V3.
2. Remove the `debtSharesMigrated == 0` check from `_gatherFromV2`. This can require more changes because debt share value of the user is used in a lot of divisions. Divide by zero's should be prevented by locking the routes that can lead to divide by zero if debt share is 0.

Resolution

Synthetix Team: Acknowledged.

L-01 | FeePool Fees Can Be Claimed With Bad C-ratio

Category	Severity	Location	Status
Unexpected Behavior	● Low	Global	Acknowledged

Description

Rewards in the FeePool can only be claimed if the user has a c-ratio smaller than the issuanceRatio as we can see here. Some of these rewards still exist in the form of unclaimed SNX which have been rolled over.

The issue lies with a user who previously could not claim in V2X due to a bad c-ratio but after migrating to V3, now has his c-ratio become 0 which then allows him to claim the rewards.

Recommendation

Be aware of this alteration in behavior, and consider informing users that they may claim these after they migrate.

Resolution

Synthetix Team: Acknowledged.

L-02 | Inflated Rewards For LegacyMarket

Category	Severity	Location	Status
Unexpected Behavior	● Low	Global	Acknowledged

Description

LiquidatorRewards.earned() calculates how much rewards a given account has earned. This earned amount is also added towards the account's collateral in Issuer.sol. There is a problem with the LegacyMarket because it’s entry has never been updated, but it holds debt shares.

This results in the earned() function returning a large amount of tokens for the LegacyMarket which will additionally inflate its collateral as well.This may cause serious problems with integrators.

Recommendation

Call updateEntry for the LegacyMarket. If you want to be extra safe, you can call it before every migration.

Resolution

Synthetix Team: Acknowledged.

L-03 | 1 Week Lock Of Free Collaterals After Migration

Category	Severity	Location	Status
DoS	● Low	Global	Acknowledged

Description

The collateral that wasn't locked before will be locked for 1 week after migrating as it can not be undelegated for a while. For example, if a staker owns $1000 collateral & $100 debt in V2 the user can transfer a part of the funds freely but is not able to do so for 1 week after migrating to V3.

Recommendation

Informs users that their funds will be locked for 1 week after migrating to V3.

Resolution

Synthetix Team: Acknowledged.

L-04 | _calculateDebtValueMigrated May Divide By 0

Category	Severity	Location	Status
Unexpected Behavior	● Low	LegacyMarket	Resolved

Description

`LegacyMarket._calculateDebtValueMigrated` divides by the second variable returned from `Issuer.allNetworksDebtInfo` which can be 0. This scenario may be reached if all synths have been burned or migrated.

Recommendation

Consider if reverting, the current behavior, is desirable.

Resolution

Synthetix Team: The issue was resolved in commit 67bf01f.

L-05 | Excessive Escrows Instantly Claimable

Category	Severity	Location	Status
Unexpected Behavior	● Low	Legacy Market	Acknowledged

Description

In case a user has more than 1000 escrows before migrating to v3, escrows from 1001th and above will be instantly claimable. This scenario is unlikely and no accounts were found to have this number of escrows. This findings serves only to document this risk.

Recommendation

Be aware of this risk.

Resolution

Synthetix Team: Acknowledged.

L-06 | Migrated Funds May Not Be Withdrawable

Category	Severity	Location	Status
Documentation	● Low	Legacy Market	Resolved

Description

When collateral is migrated to v3, it may not be withdrawable if there is a capacity locked market. In such a case, the migrated users will have to wait until the pool's collateral raises above the given market's `minimumCredit`.

Recommendation

Make sure to warn users about this behavior.

Resolution

Synthetix Team: Resolved.

L-07 | TrustedMulticallForwarder Incompatibility

Category	Severity	Location	Status
Unexpected Behavior	● Low	LegacyMarket	Acknowledged

Description

The Legacy Market should be compatible with the MulticallTrustedForwarder. This is why the contract uses ERC2771Context._msgSender() to reference the sender.

However, the functions of the market guarded with the onlyOwner modifier will not be compatible with the TrustedMulticallForwarder because OwnableStorage.onlyOwner() uses raw msg.sender.

This incompatibility is a problem - the usage of ERC2771 in the event emissions of the same functions indicate that they should accept calls from the forwarder.

Recommendation

Use ERC2771Context._msgSender() for the access control.

Resolution

Synthetix Team: Acknowledged.

L-08 | Forced Exposure To Back BFP And Spot Market

Category	Severity	Location	Status
Unexpected Behavior	● Low	Global	Acknowledged

Description

V2 Stakers are forced to back BFP positions and Spot Market via migration. Although this is intended, it should be clearly documented to users because of the inherent risks of the V3.

Some examples:
1. Vault might be liquidated, while such thing was not an expected thing in v2, now stakers taking further risks automatically.
2. Market capacities of other markets can be locked. Because of this, stakers might not be able to undelegate and withdraw their collateral back for some time.
3. Pool can go outRange in terms of `maxPoolShare` which also affects LP's.

Recommendation

Inform the users about the inherent risks they will take when they migrate.

Resolution

Synthetix Team: Acknowledged.

L-09 | distributeDebtToPools Calls Not Future Proof

Category	Severity	Location	Status
DoS	● Low	Pool.sol: 300	Acknowledged

Description

The distributeDebtToPools function is called multiple times in the codebase with a very high maxIter value: 999999999.

Therefore depending on the number of pools in the system the bumpPools function could loop up to 999999999 times.

This can lead to a block gas limit DoS in the future when pool creation is permissionless.

Recommendation

Be aware of this potential block gas limit DoS and consider reducing the maxIter value.

Resolution

Synthetix Team: We offer a public backdoor function which allows calling this specific routine with a lower limit if it becomes necessary.
https://github.com/Synthetixio/synthetix-v3/blob/main/protocol/synthetix/contracts/modules/core/MarketManagerModule.sol#L278.

L-10 | _calculateDebtValueMigrated Should Round Up

Category	Severity	Location	Status
Logical Error	● Low	LegacyMarket.sol: 354	Acknowledged

Description

The `_calculateDebtValueMigrated` function calculates the value of the migrated debt shares. This value is later used to associate it with the stakers account.

As the `_calculateDebtValueMigrated` function rounds down a small amount of debt is distributed among all accounts in the pool instead of only to the staker.

Recommendation

Round up in `_calculateDebtValueMigrated` function instead.

Resolution

Synthetix Team: Acknowledged.

L-11 | Market Weights And Monitoring

Category	Severity	Location	Status
Warning	● Low	Global	Acknowledged

Description

In the current configuration, the spartan preferred pool is expected to back the LegacyMarket, BFP Market and a Spot Market. The weights for these markets should be chosen very carefully.

Consider this:
In the case where all users want to convert their sUSD(v2) to snxUSD(v3), if the Legacy Market doesn't have enough capacity to let these conversions happen, some sUSD will won't be convertable to snxUSD.

Considering the utilization of sUSD is now very restricted (e.g exchanges are suspended), sUSD can lose its peg and there might be considerable losses for users who remain in the v2 side.

Hence it is important that weights are chosen such that it will be possible to migrate all v2 synths to v3 within expected timeframes.

Recommendation

Choose a weight for Legacy Market such that it will be possible to migrate all v2 synths to v3. Monitor migrations and be ready to adjust the weights accordingly.

Resolution

Synthetix Team: The purpose of not removing liquidity entirely after migration is because the debt on the v2x system might still increase, and someone has to take on that change, but other than that, it's pretty small.

L-12 | Missing Check In mintUsd

Category	Severity	Location	Status
Validation	● Low	IssueUSDModule.sol: 52-105	Resolved

Description

Liquidating a vault can be very profitable for the liquidator and the mintUsd function does not check if the vault is liquidatable.

This allows to bring the vault into an unhealthy state on purpose, or to get out as many funds as possible before or during a vault liquidation. Though the extent to which this is possible is limited by the issuanceRatio.

Recommendation

Revert in the mintUsd function if the vault is liquidatable.

Resolution

Synthetix Team: The issue was resolved in commit 915b19a.

L-13 | Min Delegation Amount Can Prevent Liquidations

Category	Severity	Location	Status
Logical Error	● Low	LegacyMarket.sol: 259	Acknowledged

Description

During migration, delegateCollateral is called, which has a minimum delegation amount requirement check with requireSufficientDelegation.

Hence in order for delegateCollateral to be successful, collateralMigrated should be bigger than minDelegation config of specified collateral. Any migration hence liquidation with amount less than this value will revert.

In the current configurations in Cannon, for SNX token, this amount is 100e18. This will correspond to:

With current value of SNX ($1.33)	: $133
With value from 1 months ago($2)	: $200
With value from 4 months ago($5)	: $500

Staker positions below these values won't be migratable and hence won't be liquidatable. Considering migration will last 3-12 month expectedly, and big fluctuations happened in the SNX value recently, it is not unexpected for this value to be in the upper hand.

Recommendation

requireSufficientDelegation should be bypassed during migration. Lowering the limit in general could lead to another issue as the incentives for liquidators may be reduced.

Resolution

Synthetix Team: Acknowledged.

L-14 | Stablecoin Supply Can Be Inflated

Category	Severity	Location	Status
Logical Error	● Low	LegacyMarket.sol: 165	Resolved

Description

The convertUSD function checks:
• if the given amount is not zero
• if not less than the given amount - 1 was burned

Therefore it can happen that 1 snxUSD is minted by burning 0 sUSD. For example, if the circuit breaker in the burnSynths function is triggered it will just return and zero debt was burned.

Recommendation

Revert if 0 debt was burned.

Resolution

Synthetix Team: The issue was resolved in commit 3ee0bc7.

L-15 | reportedDebt Function Does Not Revert

Category	Severity	Location	Status
Validation	● Low	LegacyMarket.sol: 115	Acknowledged

Description

The `reportedDebt` function receives the `requestedMarketId` as a parameter and checks if the given id equals the own one to ensure the correct contract/market was called.

If the check fails, the functions return 0 debt instead of reverting. This may be unexpected as other markets such as BFP revert instead of silently returning 0 debt.

Recommendation

Revert instead of returning 0.

Resolution

Synthetix Team: Acknowledged.

L-16 | Setting issuanceRatio To 1 Locks Every Token

Category	Severity	Location	Status
Documentation	● Low	Issuer.sol: 440-463	Acknowledged

Description

According to SCCP-2134: LegacyMarket V3 Migration Preparations, `issuanceRatio` will be settled to 1 wei. Settling `issuanceRatio` to 1 wei not only prevent issuance, it also locks all SNX tokens for every user that has debt.

This happens because in the `transferableSynthetixAndAnyRateIsInvalid` function, there is the following calculation:

```
uint lockedSynthetixValue = debtBalance.divideDecimalRound(getIssuanceRatio());
```

In the case of `issuanceRatio` settled to 1 wei, this locked amount will be large which will prevent all collateral transfers for all users that have debt.

Recommendation

This is the expected behavior, but be sure to document this clearly for users.

Resolution

Synthetix Team: Users should only be able to unstake completely from v2x, migrate to v3 or be liquidated after we push for stakers to migrate to v3.

L-17 | minDelegation May Prevent Migration

Category	Severity	Location	Status
Unexpected Behavior	● Low	LegacyMarket	Acknowledged

Description

When an account is being migrated, its collateral from the v2 system is delegated to the preferred pool in the v3 system. This operation will fail if the user's collateral is below the configured `minDelegationD18`.

Recommendation

`requireSufficientDelegation` should be bypassed during migration. Lowering the limit in general could lead to another issue as the incentives for liquidators may be reduced.

Resolution

Synthetix Team: Acknowledged.

L-18 | Liquidator DoS Griefing Attack

Category	Severity	Location	Status
DoS	● Low	LegacyMarket.sol: 231	Acknowledged

Description

Malicious actors can grief liquidators who try to liquidate/migrate an account by front-running the call and transferring more SNX to the wallet that will be liquidated to increase the health of the position.

As the check if the account is liquidatable is executed in the middle of the `_migrate` function the liquidator will lose a decent amount of gas if the account is not liquidatable.

Recommendation

Make liquidators aware of this issue and recommend using a private RPC.

Resolution

Synthetix Team: Acknowledged.

L-19 | Market creditCapacity Mis-accounting

Category	Severity	Location	Status
Documentation	● Low	Pool.sol: 261	Acknowledged

Description PoC

In the synthetix V3 system the creditCapacity of a market does not decrease when the market reports more debt, as shown in the first attached PoC. This is because the valuePerShareD18, which already includes the totalDebt update from the market, is the base from which the effectiveMaxShareValueD18 is decided upon in the getSystemMaxValuePerShare function.

This disagrees with the documentation of the creditCapacity here, which states “the available credit capacity for a market can be calculated by taking the total credit capacity provided to it across all pools, subtracting its amount of reported debt and its net issuance”.

Additionally, this behavior means that the result of getWithdrawableMarketUsd is misleading, as the market certainly has less available value to draw upon once the backing pool absorbs debt.

Furthermore, if a delegator were to undelegate their collateral, swap it for sUSD, and pay back an equal value of debt in the pool, the creditCapacity of the market would still go down — as the debt is effectively ignored and the collateral value has been reduced.

Finally, this yields potentially unexpected cases when multiple markets are connected to a pool. In the second PoC attached, Market A’s reportedDebt increases and as a result, Market B’s creditCapacity decreases while Market A’s creditCapacity increases at Market B’s expense.

Recommendation

The behavior described in this finding directly contradicts the documentation here, which states “the available credit capacity for a market can be calculated by taking the total credit capacity provided to it across all pools, subtracting its amount of reported debt and its net issuance”.

This is in reference to the amount of withdrawable value the market has from V3, reported by the getWithdrawableMarketUsd function, but this is not accurate due to the behavior described. This behavior has been acknowledged as expected, therefore the documentation should be updated to reflect the true behavior of the market creditCapacity rebalancing.

Resolution

Synthetix Team: Acknowledged.

Disclaimer

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 Guardian 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. Guardian’s position is that each company and individual are responsible for their own due diligence and continuous security. Guardian’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 Guardian 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.

Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

About Guardian Audits

Founded in 2022 by DeFi experts, Guardian Audits is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian Audits upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit https://guardianaudits.com

To view our audit portfolio, visit https://github.com/guardianaudits

To book an audit, message https://t.me/guardianaudits