Pendle V2 Part 1

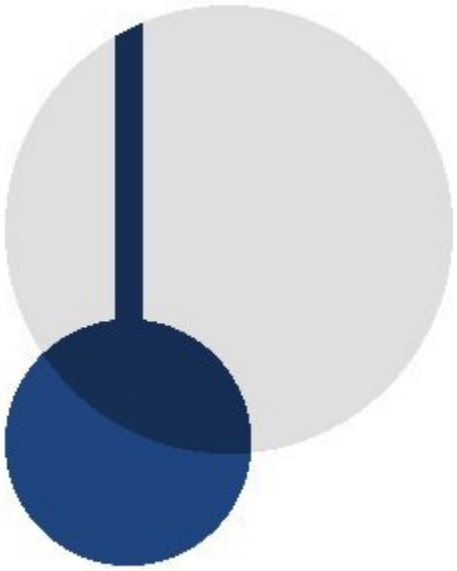


Table of Contents

Table of Contents **2**

Scope **3**

Summary of Findings **5**

Issues **6**

PEN-001: Incorrect value used for netScyOut in _callbackSwapYtForScy 6

PEN-002: mintPt should not be allowed if expiry is over 6

PEN-003: Externally pending rewards before expiry are sent to treasury if redeemRewardsPostExpiryForTreasury is done after expiry 6

PEN-004: Wrong owner parameter for redeem in PendleERC4626SCY 7

PEN-005: Lack of non zero check for netScyToAccount and netPtToAccount 7

PEN-006: IMPLIED_RATE_TIME is 360 days instead of 365 days 8

PEN-007: Lack of expiry time sanity check 8

PEN-008: Gas optimization for _doTransferOutRewards 8

PEN-009: Gas optimization for _redeemPY 8

PEN-010: Contract addresses in various SCY implementations should be underlying contracts when possible 9

PEN-011: Inconsistency between documentation and code 9

PEN-012: Unnecessary spendAllowance when redeeming SCY 10

PEN-013: joeRouter is not used in PendleJoeSwapHelperUpg 10

Scope

The scope of the audit is https://github.com/pendle-finance/pendle-core-internal-v2, with the commit hash 74b400862eff66f28b604d457e202d4f20acc387.

SCYBase.sol
SCYBaseWithRewards.sol
PendleMarket.sol
PendleMarketFactory.sol
PendleERC20.sol
PendleERC20Permit.sol
PendleRouter.sol
InterestManagerYT.sol
PendlePrincipalToken.sol
PendleYieldContractFactory.sol
PendleYieldToken.sol
ActionCallback.sol
ActionCore.sol
ActionYT.sol
ActionSCYAndPTBase.sol
ActionSCYAndPYBase.sol
ActionSCYAndYTBase.sol
CallbackHelper.sol
RewardManager.sol
RewardManagerAbstract.sol
SCYIndex.sol
SCYUtils.sol
ArrayLib.sol
MiniHelpers.sol
SSTORE2Deployer.sol
TokenHelper.sol
LogExpMath.sol
MarketApproxLib.sol
MarketMathCore.sol
Math.sol
WeekMath.sol
PendleJoeSwapHelperUpg.sol
PendleGovernanceManager.sol
PermissionsV2Upg.sol
PendleAaveV3SCY.sol
WadRayMath.sol
PendleQiTokenHelper.sol
PendleQiTokenSCY.sol
PendleERC4626SCY.sol

PendleWstEthSCY.sol
PendleYearnVaultScy.sol

Summary of Findings

In performing a security audit of Pendle V2, several issues of concern were found. For each finding, a summary of the issue is documented, along with any other finer details regarding the issue. Security recommendations are also provided where applicable.

The table below shows a breakdown of security findings found categorized by severity or risk and impact. A finding that has been reported is listed as pending, and if that finding is satisfactorily mitigated, it will be categorized as resolved.

Severity	Resolved	Unresolved	Total
Critical	0	0	0
High	0	0	0
Medium	1	0	1
Low	5	0	5
Info	7	0	7

Issues

PEN-001: Incorrect value used for netScyOut in _callbackSwapYtForScy

Severity: Medium
Status: Resolved
The check for require(netScyOut >= minScyOut) does not subtract scyOwed, thus resulting in a greater amount for netScyOut, even though the amount of SCY the user receives is netScyOut - scyOwed.

This results in an improper check for the minScyOut which the user should receive.

Recommendations
Subtract scyOwed from netScyOut before checking it in the require statement.

Resolution
Fixed by adjusting netScyOut = totalScyRedeemed - scyOwed

PEN-002: mintPt should not be allowed if expiry is over

Severity: Low
Status: Resolved
There is a lack of checking the mint time against the expiry in mintPt, thus allowing Pendle Yield Tokens to be minted even after expiry.

Recommendations
Add a check to ensure that PT cannot be minted once expired.

Resolution
A check has been added to ensure no minting after expiry. Furthermore, _isExpired is also exposed using an external view function to allow reading the expiry for a market.

PEN-003: Externally pending rewards before expiry are sent to treasury if redeemRewardsPostExpiryForTreasury is done after expiry

Severity: Low
Status: Resolved
In PendleYieldToken, if there are externally pending rewards which have not been claimed before expiry is up, and post-expiry, when redeemRewardsPostExpiryForTreasury, those pending rewards before the expiry would go to the treasury. This could result in lesser rewards than the user is entitled to.

Recommendations
Consider giving rewards up to the first interaction with PendleYieldToken's expiry to the user.

Resolution
The design has been changed to call _setPostExpiryData which is called on every external interaction with PendleYieldToken after expiry. Rewards up to this point will belong to users, while any reward after the setting of post expiry data will belong to the treasury.

PEN-004: Wrong owner parameter for redeem in PendleERC4626SCY

Severity: Low
Status: Resolved
For ERC4626 redeem, the 3rd param is owner, which is the address which the ERC4626 receipt token will be burnt from. The address should be address(this) (SCY), not msg.sender (the user)

```
amountTokenOut = IERC4626(yieldToken).redeem( // ok correct since based on shares
            amountSharesToRedeem, //ok
            address(this), //receiver is SCY
            msg.sender //@audit owner should be address(this)
        );
```

This will revert and users will be unable to withdraw the underlying asset, only as ERC4626.

Recommendations
Change the owner from msg.sender to address(this)

Resolution
The change of the owner parameter has been made.

PEN-005: Lack of non zero check for netScyToAccount and netPtToAccount

Severity: Low
Status: Resolved
In removeLiquidityCore, there is a check that lpToRemove is non-zero, but no check that scyToAccount and netPtToAccount are non-zero. If the value is somehow zero, due to the denominator being greater than the numerator during the calculation, removeLiquidity will cause 0 tokens to be transferred when removing the liquidity, even if lpToRemove is non-zero.

Recommendations
Add a check require(netScyToAccount > 0 || netPtToAccount > 0);

Resolution
The recommended check has been added.

PEN-006: IMPLIED_RATE_TIME is 360 days instead of 365 days

Severity: Low
Status: Resolved
IMPLIED_RATE_TIME is supposed to be 360 days instead of 365 days (1 year)

Recommendations
Change the value to 365 days.

Resolution
The change to 365 days has been made.

PEN-007: Lack of expiry time sanity check

Severity: Info
Status: Acknowledged
Consider adding an upper limit cap for expiry in createYieldContract. This can prevent human error (e.g one too many zeros) resulting in an expiry time that will never be arrived at in a reasonable time.

Recommendations
Add a reasonable maximum limit for the expiry time.

Resolution
The expiry time has been changed from uint256 to uint32.

PEN-008: Gas optimization for _doTransferOutRewards

Severity: Info
Status: Resolved
In RewardManager's _doTransferOutRewards, rewardState[token].lastBalance can be decremented only if rewardAmounts[i] is non-zero. Otherwise, zero would be subtracted from lastBalance, resulting in no change anyway.

Recommendations
rewardState[token].lastBalance is only decremented if rewardAmounts[i] is non-zero.

Resolution

PEN-009: Gas optimization for _redeemPY

Severity: Low
Status: Resolved

In PendleYieldToken's _redeemPY, in the for loop which send out SCY to receivers, if totalAmountRemains is 0 after subtraction, the loop can break early as subsequent Math.min(totalAmountRemains, amounts[i]); will return 0, even if there are still receivers.

Recommendations
Break out of the for loop if totalAmountRemains is 0.

Resolution
The for loop now breaks early if totalAmountRemains is 0.

PEN-010: Contract addresses in various SCY implementations should be underlying contracts when possible

Severity: Info
Status: Resolved
For SCY contracts where multiple required contracts such as the underlying and receipt tokens are required, if they can be obtained from a single contract such as the the staking or lending contract, that should be done instead of as parameters.

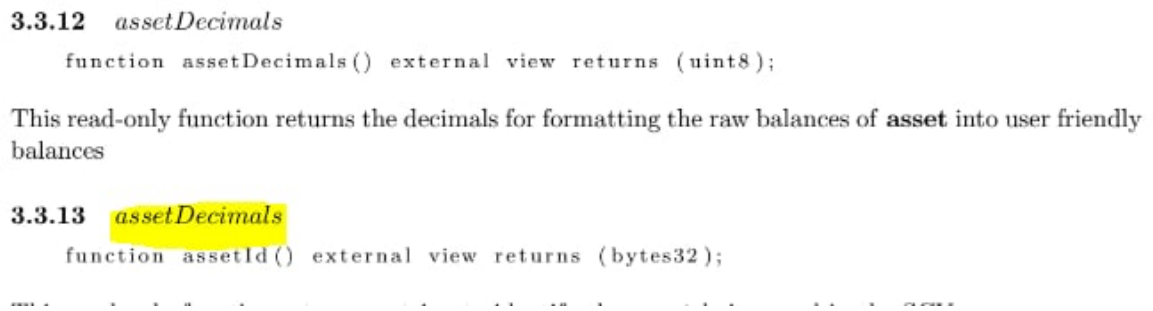
For example, for PendleBtrflySCY, xBTRFLY and BTRFLY should be obtained from the wxBTRFLY contract.

Recommendations
Review the SCY implementation contracts to consider what can be removed from the constructor parameters and obtained from contract calls instead.

Resolution
Unnecessary constructor parameters have been removed from the various SCY tokens.

PEN-011: Inconsistency between documentation and code

Severity: Info
Status: Resolved
assetId() is present in the docs, but not in the contract code.



3.3.12 *assetDecimals*

```
function assetDecimals() external view returns (uint8);
```

This read-only function returns the decimals for formatting the raw balances of **asset** into user friendly balances

3.3.13 *assetDecimals*

```
function assetId() external view returns (bytes32);
```

Recommendations
Update the document based on the contract code.

Resolution
The documentation has been updated to reflect the code.

PEN-012: Unnecessary spendAllowance when redeeming SCY

Severity: Info
Status: Resolved
In redeem, if amountSharesToPull is non-zero, as the SCY token is being transferred to the SCY token contract itself, there is no need to call _spendAllowance. If you look at other tokens like CTokens or Staked tokens like SAVAX, if they are transferring the receipt token to the token contract itself, they can access the _transfer or _burn internal function directly.

The current implementation would mean that the user would need to grant allowance of the SCY token to the SCY token address itself.

Recommendations
Remove spendAllowance in redeem.

Resolution
spendAllowance has been removed.

PEN-013: joeRouter is not used in PendleJoeSwapHelperUpg

Severity: Info
Status: Resolved
joeRouter does not seem to be used in any of the code even though declared in the constructor.

Recommendations
Remove joeRouter if unnecessary.

Resolution
joeRouter has been removed.