2024-03-14 - Stake.Link Mitigation Review

Cyfrin

1. Mitigations in Commit **dc86d4ae58024d844f875a23eba4a6d393c61b51** (Core Contest Findings)

FINDINGS	Severity	STAKE.LINK Comments	Cyfrin Mitigation Review
H-01. A user can steal an already transfered and bridged reSDL lock because of approval	HIGH	Fixed	Mitigated *Token approvals explicitly deleted in handleOutgoingRESDL on primary and secondary chains*
• H-02. Not Update Rewards in `handleIncomingUpdate` Function of `SDLPoolPrimary` Leads to Incorrect Reward Calculations	HIGH	Fixed	Mitigated. *Rewards updating on incoming update*
• M-01. A user can lose funds in `sdlPoolSecondary` if tries to add more sdl tokens to a lock that has been queued to be completely withdrawn	MEDIUM	Fixed	Mitigated. *Reverts if lock.amount == 0*
• M-02. Attacker can exploit lock update logic on secondary chains to increase the amount of rewards sent to a specific secondary chain	MEDIUM	Fixed	Mitigated. *Extra condition to chk boost amount == 0*
• L-01. SINGLE STEP OWNERSHIP TRANSFER PROCESS	LOW	Acknowledged	Acknowledged
• L-02. CCIP router address cannot be updated	LOW	Fixed	Mitigated. *setRouter introduced in ccipReceiver*

• L-03. Accidental `renounceOwnership()` call can disrupt key operations in multiple contracts.	LOW	Acknowledged	Acknowledged
• L-04. Insufficient Gas Limit Specification for Cross-Chain Transfers in _buildCCIPMessage() method.	LOW	Fixed	Mitigated. *0 default is used*
• L-05. No validation for `_amount` in migrate function	LOW	Fixed	Mitigated.
• L-06. Lack of storage gap in SDLPool.sol can lead to upgrade storage slot collision.	LOW	Fixed	Mitigated. *Introduced storage gap. Openzeppelin recommended slots are 50 → current implementation uses 3.*
• L-07. Fee Calculation inconsistency in WrappedTokenBridge	LOW	Fixed	Mitigated, *1000 ether is replaced by actual amount*
• L-08. WrappedTokenBridge#recoverTokens will drain the whole token balance	LOW	Fixed	Mitigated. *recovery amount is now passed as input*
• L-09. Can lock Fund for 1 sec and unlock in same transaction to gain profit	LOW	Fixed	Mitigated *min lock duration introduced*
• L-10. No Check for Transferring to Self	LOW	Fixed	Mitigated *Reverts when from == to*
• L-11. Audit Report for SDLPool.sol - Scalability Concern	LOW	Acknowledged	Acknowledged.
• L-12. Updates from the `secondary pool` to the `primary_pool` may not	LOW	Acknowledged	Acknowledged. Protocol ensures

be sent because there are `no rewards` for the secondary pool			
• L-13. Single strategy failure blocks global reward distribution	LOW	Acknowledged	Acknowledged

2. Mitigations/Updates in Commit: 9ab52b273bb809e4bd566c6673367abb3d0b7353 (Related to dynamic gas limits)

CHANGES	**Cyfrin Review/Comments**
1. Replace extraArgs with gasLimit	Note: By replacing `extraArgs` with `gasLimit` , we have made the codebase less compatible to future CCIP upgrades. This is OK for now from a security standpoint and considering that the contract itself is upgradeable.
2. Queue ccipUpdates on primary chain	OK
3. Execute queued ccipUpdates using `onlyUpdateInitiator`	OK
4. Removed extraArgs and shouldUpdated. Added `timeOfLastUpdate` and `minTimeBetweenUpdates`	Note: Since these contracts are not deployed yet on mainnet, this is fine. Please be mindful of storage collisions when removing and adding new storage variables or swapping their order in upgradeable contracts
4. Added `_minTimeBetweenUpdates` and `timeOfLastUpdate` to control time between updates → onlyOwner change	OK
5. `executeUpdate` on secondary chain is now access controlled → can only be called by `onlyUpdateInitiator`	**ISSUE**: This earlier logic to execute update had no access control, ie. anyone could call an update. Now, this can only be called by `onlyUpdateInitiator` Note that when L2 sequencers are down, this new implementation prevents anyone from running `executeUpdate` - which means that these updates are not finalized on mainnet → which means that any call to `distributeRewards` does not include the pending lock ID's. For arbitrum specifically, an alias address needs to be supported to force include the update even when the

	sequencer is down. Refer to this issue below for more details https://solodit.xyz/issues/m-8-operator-is-blocked-when-sequencer-is-down-on-arbitrum-sherlock-none-index-git

Minor Fixes Recommended

#	Issue	Detail
1	Incorrect Natspec for SDLPoolCCIPControllerPrimary	Wrong title → change to " SDL Pool CCIP Controller Primary"