SB SECURITY

Omnichain (B88, E280, H420, S88) Security Review

March 17, 2025

Contents

1. About SBSecurity

SBSecurity is a duo of skilled smart contract security researchers. Based on the audits conducted and numerous vulnerabilities reported, we strive to provide the absolute best security service and client satisfaction. While it's understood that 100% security and bug-free code cannot be guaranteed by anyone, we are committed to giving our utmost to provide the best possible outcome for you and your product.

Book a Security Review with us at sbsecurity.net or reach out on Twitter @Slavcheww.

2. Disclaimer

A smart contract security review can only show the presence of vulnerabilities **but not their absence**. Audits are a time, resource, and expertise-bound effort where skilled technicians evaluate the codebase and their dependencies using various techniques to find as many flaws as possible and suggest security-related improvements. We as a company stand behind our brand and the level of service that is provided but also recommend subsequent security reviews, on-chain monitoring, and high whitehat incentivization.

3. Risk classification

	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

3.1. Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - leads to a moderate loss of assets in the protocol or some disruption of the protocol's functionality.
- **Low** - funds are not at risk.

3.2. Likelihood

- **High** - almost **certain** to happen, easy to perform, or highly incentivized.
- **Medium** - only **conditionally possible**, but still relatively likely.
- **Low** - requires specific state or **little-to-no incentive**.

3.3. Action required for severity levels

- High - **Must** fix (before deployment if not already deployed).
- Medium - **Should** fix.
- Low - **Could** fix.

4. Executive Summary

Overview

Project	B88, E280, H420, S88
Repository	Private
Commit Hash	- B88: 8097d214d4d4789715076110b56a91c9349e88af - E280: 4aea66390e203f23befcc6f89a19b9fd18fe7ade - H420: 90956dcdfb23dc46830eba42a95673f64278ca9d - S88: ecb6097994a3953a309196a83115782f2b9b142e
Resolution	- B88: 5b65d99808fd9dc9b92d008c7ca2191c039976da - E280: 8894ce4f54bd89d0dd670d79c5316a273a81d138 - H420: b71ce673e1f2df160d361189d11dfc0b1cd9d3d2 - S88: 2e3ddcdfa58ed6c70b5eabdbb79b9859f17cdcb6
Timeline	March 7, 2025 - March 17, 2025

Scope

B88/
E280/
H420/
S88/

Issues Found

Critical Risk	0
High Risk	0
Medium Risk	0
Low/Info Risk	11

5. Findings

5.1. Low/Info severity

5.1.1. For manual Lp deployments, anyone can frontrun and create the pair with skewed prices

Severity: Low Risk

Description: Anyone with either of the new OFT tokens (assuming the minting is enabled before LPs are deployed) can create the pair themselves at whatever ratio they want. Even more, they can inflate it just enough to make the initially minted OFT tokens to the `_lpDeployer` insufficient to set the desired price.

Recommendation: Consider adding a check in the `_update` that will prevent transfers until toggled, this will allow you to safely perform all the setups.

Resolution: Acknowledged

5.1.2. Loss of funds if the bridge message's amount get rounded

Severity: Low Risk

Description: LayerZero has a restriction that will round the amount bridged, if `(amount / decimalConversionRate)` is greater than `uint64.max`, it happens in the `_buildMsgAndOptions`, where the `message` is created:

```
(message, hasCompose) = OFTMsgCodec.encode(
        _sendParam.to,
        _toSD(_amountLD),
        // @dev Must be include a non empty bytes if you want to compose, EVEN if you dont need it on the
remote.
        // EVEN if you dont require an arbitrary payload to be sent... eg. '0x01'
        _sendParam.composeMsg
    );

function _toSD(uint256 _amountLD) internal view virtual returns (uint64 amountSD) {
    return uint64(_amountLD / decimalConversionRate);
}
```

Due to the casting in the `_toSD`, it will perform silent overflow and will round the number.

With the current prices of E280, it will be an unreasonably big amount (~1 million USD) but if it happens, only 1% of the funds will arrive at the destination chains, and the rest 99% will be burned forever.

Recommendation: In the `bridge` functions you can limit the amount that is bridged, divided by the `decimalsConversionRate` to a max of uint64.

Resolution: Fixed

5.1.3. Informational issues and code suggestions

Severity: Low Risk

Description:

1. `capPerSwapScale` in S88 contracts should be lower, since now `888_888_888` ether in terms of SCALE, which is 9 decimal places, is over $115 million.

2. Since the bridge amount is rounded to 1e12 due to the LayerZero local/remote decimal, the incentive is applied to the entire balance, but the entire balance will not be used. Either change the incentive or make it based on the rounded amount.

3. The minimum threshold for bridging will be effective, eliminating cases where users bridge small amounts.

4. Last calls aren't set to `block.ts` in constructors, allowing swaps/injections immediately. Consider setting last call variables in the constructor.

5. None of the setter functions in the 4 repos emit events.

6. `B88BuyBurn::burnTokens` uses wrong name for the variables (`h420`)

7. In E280.sol and S88.sol - `_applyTax()` on Base replace `100_00` with `BPS_BASE`.

8. S88 in `S88LPDepositor` is not used.

9. `IncompatibePair` error in `E280LPManager` has a typo, should be IncompatibLe.

Resolution: Fixed - 2, 6, 8, 9