SB SECURITY

E280 BNB

Security Review

June 31, 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	E280 BNB
Repository	Private
Commit Hash	6816f14fd2241606e0fba502eb92428cf92b26d8
Resolution	25bf8c97236e496a39cde405302af20f76b6d43f
Timeline	June 17, 2025

Scope

E280ProtocolLP.sol

Issues Found

Critical Risk	0
High Risk	1
Medium Risk	1
Low/Info Risk	0

5. Findings

5.1. High severity

5.1.1.1. BuyBurn won't work for V3 pairs due to wrong function signature (Out of scope)

Severity: High Risk

Description: In the BSC version of the `UniswapV3Router`, there's no deadline parameter, but the V3 swaps in the `E280BuyBurn` contract use the wrong `ExactInputParams` and `ExactInputSingleParams` structs that have `deadline`. This will cause all the V3 swaps to revert, since the non-existent function signature will be created.

```
function _handleV3Swap(address token, uint256 amountIn, uint256 minAmountOut, uint256 deadline) internal {
    IERC20(token).safeIncreaseAllowance(UNISWAP_V3_ROUTER, amountIn);
    if (isMultihopSwap[token]) {
        ISwapRouter.ExactInputParams memory params = ISwapRouter.ExactInputParams({
            path: multihopSwapOptionsV3[token],
            recipient: address(this),
            deadline: deadline,//ISSUE: there's no deadline https://bscscan.com/address/
0xb971ef87ede563556b2ed4b1c0b0019111dd85d2#code in InputSingle as well
            amountIn: amountIn,
            amountOutMinimum: minAmountOut
        });
        ISwapRouter(UNISWAP_V3_ROUTER).exactInput(params);
    } else {
        SingleSwapOptionsV3 memory options = swapOptionsV3[token];
        ISwapRouter.ExactInputSingleParams memory params = ISwapRouter.ExactInputSingleParams({
            tokenIn: token,
            tokenOut: options.tokenOut,
            fee: options.fee,
            recipient: address(this),
            deadline: deadline,
            amountIn: amountIn,
            amountOutMinimum: minAmountOut,
            sqrtPriceLimitX96: 0
        });
        ISwapRouter(UNISWAP_V3_ROUTER).exactInputSingle(params);
    }
}
```

Recommendation: Copy the structs directly from the BSC contract.

Resolution: Fixed

5.2. Medium severity

5.2.1. E280 will be locked in the `E280ProtocolLP` after sunset

Severity: Medium Risk

Description: `E280` will continuously enter the LP contract, in order to be swapped for `X28, BTCB and WBNB`, but when the contract is being sunset, through `sunsetInjector`, the accumulated but not swapped E280 won't be transferred.

```
function sunsetInjector(
    uint256 minAmountOutBtcb,
    uint256 minAmountOutBnb1,
    uint256 minAmountOutX28,
    uint256 minAmountOutBnb2,
    uint256 deadline
) external onlyOwner {
    if (injectorSunset) revert Prohibited();
    _removeLiquidity(BTCB_LP_TOKEN_ID, minAmountOutBtcb, minAmountOutBnb1, deadline);
    _removeLiquidity(X28_LP_TOKEN_ID, minAmountOutX28, minAmountOutBnb2, deadline); // CHECK TOKEN 0

    IERC20(BTCB).safeTransfer(E280_BUY_BURN, IERC20(BTCB).balanceOf(address(this)));
    IERC20(X28).safeTransfer(E280_BUY_BURN, IERC20(X28).balanceOf(address(this)));
    IERC20(WBNB).safeTransfer(E280_BUY_BURN, IERC20(WBNB).balanceOf(address(this)));

    injectorSunset = true;
}
```

Furthermore, `E280TaxDistributor` isn't notified for the sunset and will continue to supply `E280` tokens back to the injector. These tokens will also remain stuck in the `E280ProtocolLP` contract.

```
function distribute() external nonReentrant {
    IERC20 e280 = IERC20(E280);
    uint256 balance = e280.balanceOf(address(this));
    if (balance == 0) revert InsufficientBalance();

    balance = _processIncentiveFee(E280, balance, incentiveFeeBps);
    uint256 lpInjectorAmount = _applyBps(balance, LP_INJECTOR_TAX_ALLOCATION);
    uint256 lpPromoAmount = _applyBps(balance, LP_PROMO_TAX_ALLOCATION);
    uint256 rewardsAmount = balance - lpInjectorAmount - lpPromoAmount;

    e280.safeTransfer(E280_PROTOCOL_LP, lpInjectorAmount);
    e280.safeTransfer(E280_LP_PROMO, lpPromoAmount);
    e280.safeTransfer(E280_REWARDS, rewardsAmount);

    emit Distribution();
}
```

Recommendation:

1. in `sunsetInjector` transfer the `E280` tokens as well

2. add logic to notify the `E280TaxDistributor` contract when sunset happens to stop forwarding `E280` tokens to the `E280ProtocolLP`.

Resolution: Fixed