Orbs Spot Audit Report

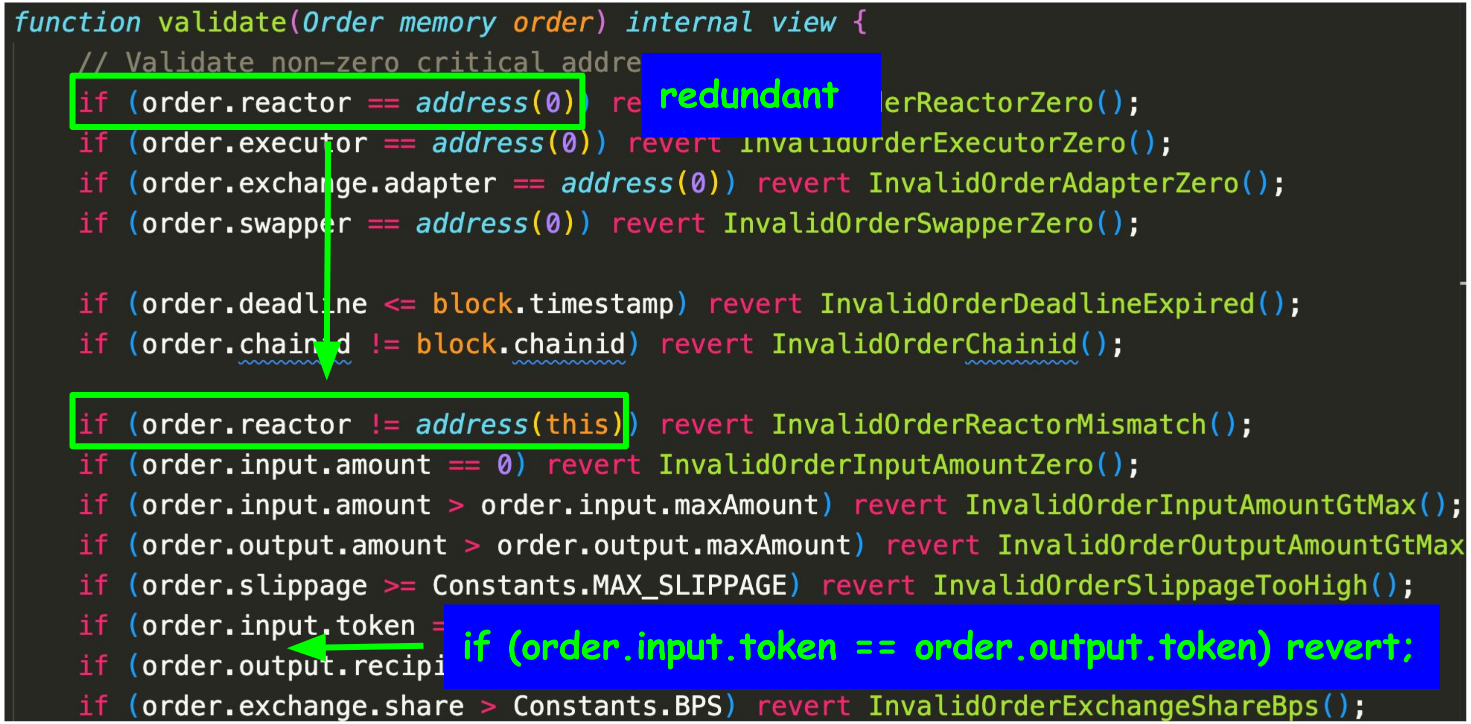
By AstraSec Team

September 23, 2025

Scope

- https://github.com/orbs-network/spot.git
- Commit ID: 542e1b3

N1: Improved Logic of OrderValidationLib::validate()

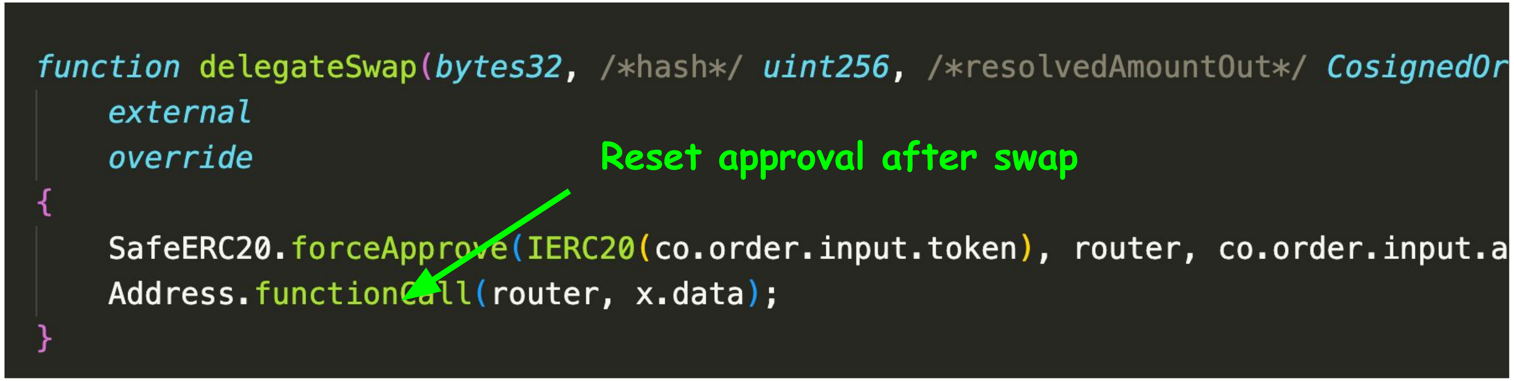


```
function validate(Order memory order) internal view {
    // Validate non-zero critical addre
    if (order.reactor == address(0)) re                    erReactorZero();
    if (order.executor == address(0)) revert InvalidOrderExecutorZero();
    if (order.exchange.adapter == address(0)) revert InvalidOrderAdapterZero();
    if (order.swapper == address(0)) revert InvalidOrderSwapperZero();

    if (order.deadline <= block.timestamp) revert InvalidOrderDeadlineExpired();
    if (order.chainid != block.chainid) revert InvalidOrderChainid();

    if (order.reactor != address(this)) revert InvalidOrderReactorMismatch();
    if (order.input.amount == 0) revert InvalidOrderInputAmountZero();
    if (order.input.amount > order.input.maxAmount) revert InvalidOrderInputAmountGtMax();
    if (order.output.amount > order.output.maxAmount) revert InvalidOrderOutputAmountGtMax
    if (order.slippage >= Constants.MAX_SLIPPAGE) revert InvalidOrderSlippageTooHigh();
    if (order.input.token =
    if (order.output.recipi
    if (order.exchange.share > Constants.BPS) revert InvalidOrderExchangeShareBps();
```

redundant

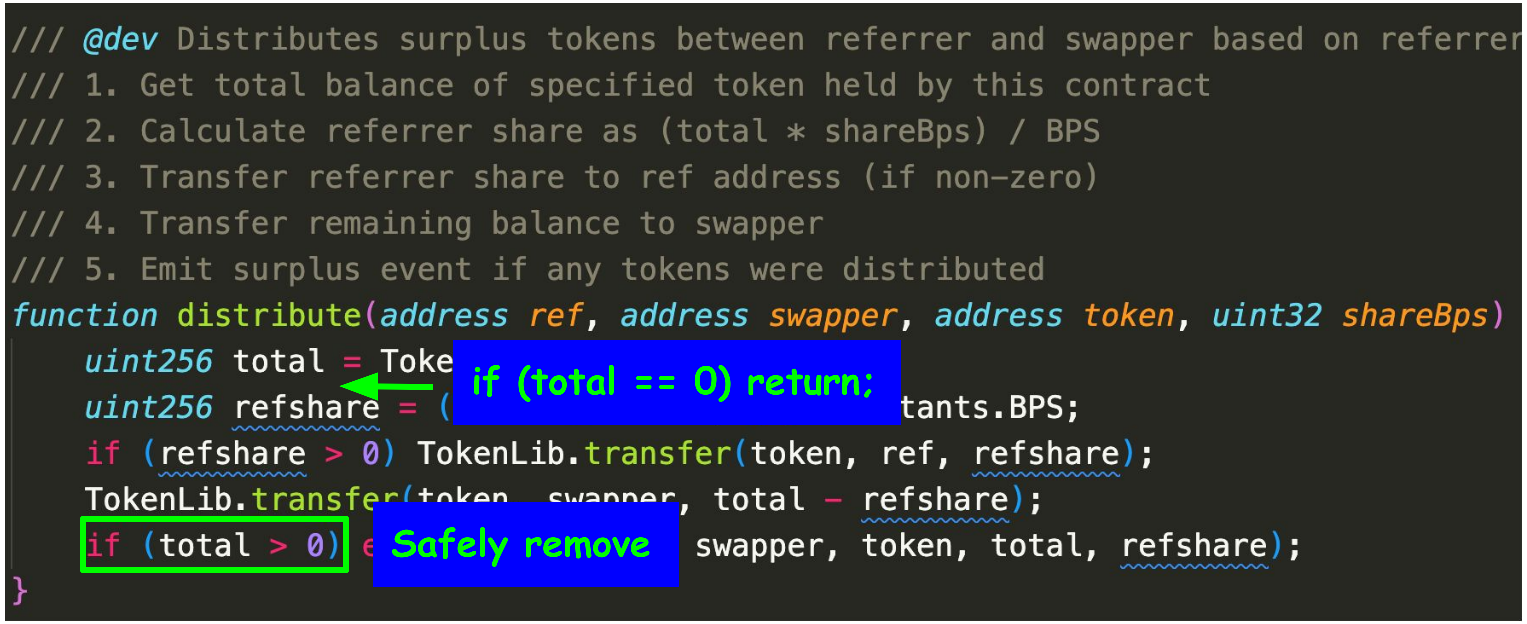
if (order.input.token == order.output.token) revert;

N2: Improved Logic of DefaultDexAdapter::delegateSwap()



```
function delegateSwap(bytes32, /*hash*/ uint256, /*resolvedAmountOut*/ CosignedOr
    external
    override
{
    SafeERC20.forceApprove(IERC20(co.order.input.token), router, co.order.input.a
    Address.functionCall(router, x.data);
}
```

N3: Improved Logic of SurplusLib::distribute()



```
/// @dev Distributes surplus tokens between referrer and swapper based on referrer
/// 1. Get total balance of specified token held by this contract
/// 2. Calculate referrer share as (total * shareBps) / BPS
/// 3. Transfer referrer share to ref address (if non-zero)
/// 4. Transfer remaining balance to swapper
/// 5. Emit surplus event if any tokens were distributed
function distribute(address ref, address swapper, address token, uint32 shareBps)
    uint256 total = Toke
    uint256 refshare = (                           tants.BPS;
    if (refshare > 0) TokenLib.transfer(token, ref, refshare);
    TokenLib.transfer(token, swapper, total - refshare);
    if (total > 0) e                   swapper, token, total, refshare);
}
```