AstraSec

Orbs TWAP
Security Audit Report

April 5, 2025

Contents

1 Introduction

1.1 About Orbs TWAP

The **Orbs TWAP** protocol implements a TWAP Order mechanism (supporting both Limit Order and Market Order) with support for partial fills. It is designed to reduce the market impact of large orders and mitigate volatility by breaking trades into smaller portions and executing them at regular intervals over a specified period.

1.2 Source Code

The following source code was reviewed during the audit:

- https://github.com/orbs-network/twap/compare/a506be1..master
- CommitID: 127f94c

2 Overall Assessment

This report has been compiled to identify issues and vulnerabilities within the Orbs TWAP protocol. Throughout this audit, we identified a total of 2 issues spanning various severity levels. By employing auxiliary tool techniques to supplement our thorough manual code review, we have discovered the following findings.

Severity	Count	Acknowledged	Won't Do	Addressed
Critical	–	–	–	–
High	–	–	–	–
Medium	–	–	–	–
Low	–	–	–	–
Informational	2	2	–	–
Undetermined	–	–	–	–

3 Vulnerability Summary

3.1 Overview

Click on an issue to jump to it, or scroll down to see them all.

3.2 Security Level Reference

In web3 smart contract audits, vulnerabilities are typically classified into different severity levels based on the potential impact they can have on the security and functionality of the contract. Here are the definitions for critical-severity, high-severity, medium-severity, and low-severity vulnerabilities:

Severity	Acknowledged
C-X (Critical)	A severe security flaw with immediate and significant negative consequences. It poses high risks, such as unauthorized access, financial losses, or complete disruption of functionality. Requires immediate attention and remediation.
H-X (High)	Significant security issues that can lead to substantial risks. Although not as severe as critical vulnerabilities, they can still result in unauthorized access, manipulation of contract state, or financial losses. Prompt remediation is necessary.
M-X (Medium)	Moderately impactful security weaknesses that require attention and re-mediation. They may lead to limited unauthorized access, minor financial losses, or potential disruptions to functionality.
L-X (Low)	Minor security issues with limited impact. While they may not pose significant risks, it is still recommended to address them to maintain a robust and secure smart contract.
I-X (Informational)	Warnings and things to keep in mind when operating the protocol. No immediate action required.
U-X (Undetermined)	Identified security flaw requiring further investigation. Severity and impact need to be determined. Additional assessment and analysis are necessary.

3.3 Vulnerability Details

3.3.1 [I-1] Inconsistent Naming Conventions

Target	Category	IMPACT	LIKELIHOOD	STATUS
ExchangeV2.sol	Coding Practice	NA	NA	Acknowledged

During the audit, we identify inconsistent naming conventions in the swap() functions of ExchangeV2 and ParaswapExchange, specifically with parameters and variables such as minOut, swapData, src, and dst. Inconsistent naming can hinder code readability and maintainability. To enhance clarity and ensure consistency, we recommend standardizing naming conventions across all contracts.

twap-master - ExchangeV2.sol

```
37 function swap(
38     address _srcToken,
39     address _dstToken,
40     uint256 amountIn,
41     uint256 minOut,
42     bytes calldata,
43     bytes calldata bidData,
44     address taker
45 ) public {
46     if (!allowed[taker]) revert TakerNotAllowed(taker);
47
48     (, bytes memory swapData) = decode(bidData);
49     IERC20 src = IERC20(_srcToken);
50     IERC20 dst = IERC20(_dstToken);
51
52     ...
53 }
```

Remediation Adopt a consistent naming convention across all contracts.

3.3.2 [I-2] Incompatibility with Non-Standard ERC20 Tokens

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
Multiple Contracts	Business Logic	NA	NA	Acknowledged

In the current implementation, safeApprove() and safeIncreaseAllowance() are used for token approvals, but they do not adequately handle non-standard ERC20 tokens, potential leading to unexpected reverts:

- ➔ The call to safeApprove() will revert if the existing allowance is non-zero, due to its front-running protection mechanism.
- ➔ Tokens such as USDT prohibit non-zero allowance changes, causing safeIncreaseAllowance() to fail when trying to modify an existing allowance.

twap-master - Taker.sol

```
36 function fill(uint64 id, address fee, uint256 dstSenderAmount, address feeSwapExchange, bytes calldata feeSwapData)
37     external
38     onlyAllowed
39 {
40     // fill
41     twap.fill(id);
42     OrderLib.Order memory o = twap.order(id);
43
44     // swap to gas
45     bool swapGas = feeSwapExchange != address(0) && o.ask.dstToken != twap.iweth() && o.ask.dstToken != address(0);
46     if (swapGas) {
47         ERC20(o.ask.dstToken).safeApprove(feeSwapExchange, dstSenderAmount);
48         IExchange(feeSwapExchange).swap(
49             o.ask.dstToken, twap.iweth(), dstSenderAmount, 0, o.ask.data, feeSwapData, address(this)
50         );
51     }
52
53     ...
54 }
```

Remediation Replace safeApprove()/safeIncreaseAllowance() with forceApprove().

4 Appendix

4.1 About AstraSec

AstraSec is a blockchain security company that serves to provide high-quality auditing services for blockchain-based protocols. With a team of blockchain specialists, AstraSec maintains a strong commitment to excellence and client satisfaction. The audit team members have extensive audit experience for various famous DeFi projects. AstraSec's comprehensive approach and deep blockchain understanding make it a trusted partner for the clients.

4.2 Disclaimer

The information provided in this audit report is for reference only and does not constitute any legal, financial, or investment advice. Any views, suggestions, or conclusions in the audit report are based on the limited information and conditions obtained during the audit process and may be subject to unknown risks and uncertainties. While we make every effort to ensure the accuracy and completeness of the audit report, we are not responsible for any errors or omissions in the report.

We recommend users to carefully consider the information in the audit report based on their own independent judgment and professional advice before making any decisions. We are not responsible for the consequences of the use of the audit report, including but not limited to any losses or damages resulting from reliance on the audit report.

This audit report is for reference only and should not be considered a substitute for legal documents or contracts.

4.3 Contact

Phone	+86 156 0639 2692
Email	contact@astrasec.ai
Twitter	https://twitter.com/AstraSecAI