



SMART CONTRACT SECURITY AUDIT

Oxiswap

DEX Contracts

CLIENT

Oxiswap

PLATFORM

Fuel

LANGUAGE

Sway

REVIEW

October 2024

DOCUMENT

Document information

Project	Oxiswap, DEX Contracts
Client	Oxiswap
Project URL	https://www.Oxiswap.com
Platform	Fuel
Language	Sway
Fork of	Uniswap v2
Repository	oxiswap/oxiswap-crypto-contract
Commit	Not recorded in the original report
Methods	Manual Analysis, Functional Testing, Automated Testing
First review	9 to 14 October 2024
Updated code received	9 October 2024

Document history

VERSION	DATE	CHANGE
1.0	October 2024	Original report issued View original report
1.1	October 2026	Reformatted into the Safe Edges standard template. Findings unchanged

This report was reformatted into the Safe Edges standard template in October 2026. Findings, severities, recommendations and code references are unchanged from the original report, which is preserved unchanged in the [Safe Edges Audit Portfolio on GitHub](#).

CONTENTS

Contents

01	Executive summary	3
02	Scope	4
03	Methodology	5
04	Severity and status	6
05	Findings summary	7
06	Detailed findings	8
	I-01 Withdrawal Function Can't Handle Zero Value Deposits	8
	I-02 ZERO_B256 deprecated	8
A	Appendix: vulnerability checklist	9
	Disclaimer	10

Executive summary

About Oxiswap

Oxiswap is a lightning-fast decentralized exchange (DEX) built on the Fuel network, designed for seamless trading, earning, and building. By leveraging the power of the Sway language and the UTXO model, Oxiswap offers an advanced decentralized trading experience with enhanced speed and security.

Summary of the review

Safe Edges reviewed the Oxiswap contracts, a Uniswap v2 fork written in Sway for the Fuel network, between 9 and 14 October 2024. Two Informational issues were identified, and the Oxiswap team fixed both.

Findings by severity



Findings by status

STATUS	ISSUES
● Resolved	2

Scope

Repository	oxiswap/oxiswap-crypto-contract
Commit	Not recorded in the original report
Platform	Fuel
Language	Sway

Only the code at the commit above was reviewed. Changes made after this commit are not covered by this report.

Methodology

Methods recorded for this engagement: Manual Analysis, Functional Testing, Automated Testing.

Manual review

Auditors read the in-scope code line by line, tracing control flow, state changes and trust boundaries against the intended behaviour described by the client.

Threat modelling

We map privileged roles, external dependencies and value flows, then test the assumptions the code makes about each of them.

Functional and automated testing

Where tooling exists for the language and chain, we run static analysis and targeted tests to confirm suspected issues and check edge cases.

Fix review

After the client addresses the findings, we review the updated code and record the final status of each issue.

Severity and status

Severity levels

SEVERITY	DEFINITION
Critical	Direct loss or permanent lock of funds, or full compromise of the protocol, with few or no preconditions. Must be fixed before deployment.
High	Significant loss of funds, broken core functionality or manipulation of critical state under realistic conditions. Should be fixed before deployment.
Medium	Limited loss or degraded functionality, or issues that need specific conditions or privileged access to exploit. Should be fixed.
Low	Minor impact, low likelihood, or deviations from best practice. Fix when practical.
Informational	Code quality, documentation, gas and style observations with no direct security impact.

Status definitions

STATUS	DEFINITION
Resolved	The issue was fixed and the fix was verified.
Partially resolved	The risk was reduced but not fully removed.
Acknowledged	The client accepted the issue without changing the code.
Open	The issue was not addressed at the time of the report.
Not recorded	The original report did not record a resolution status for this issue.

Findings summary

ID	TITLE	SEVERITY	STATUS
I-01	Withdrawal Function Can't Handle Zero Value Deposits	Informational	● Resolved
I-02	ZERO_B256 deprecated	Informational	● Resolved

Detailed findings

I-01 Withdrawal Function Can't Handle Zero Value Deposits

Severity **Informational** Status **Resolved**

DESCRIPTION

In the current implementation of the withdraw function, it stores the deposit value in the contract's storage even if the deposit becomes zero after a withdrawal. It continues to store zero values in the contract's storage,

```
let new_deposit = old_deposit - amount;  
storage.deposits.insert((sender, asset_id), new_deposit);
```

Reference: <https://github.com/oxiswap/oxiswap-crypto-contract/blob/main/crypto-router/src/main.sw#L108C1-L127C1>

RECOMMENDATION

Modify the function to remove storage entries when the new balance is zero

I-02 ZERO_B256 deprecated

Severity **Informational** Status **Resolved**

DESCRIPTION

Use of the ZERO_B256 is deprecated as per the updated Sway language standards

Reference: https://fuellabs.github.io/sway/master/std/constants/constant.ZERO_B256.html

APPENDIX A

Vulnerability checklist

Vulnerability classes checked during this review, as listed in the original report.

✓ Arbitrary write to storage	✓ Upgradeable safety
✓ Centralization of control	✓ Using inline assembly
✓ Ether theft	✓ Style guide violation
✓ Improper or missing events	✓ Parallel Execution safety
✓ Logical issues and flaws	✓ UTXO Model Verification
✓ Arithmetic Computations Correctness	✓ FuelVM Opcodes
✓ Race conditions/front running	✓ Cross-Chain Interactions
✓ Re-entrancy	✓ Modular Design
✓ Malicious libraries	✓ Access Control Vulnerabilities
✓ Address hardcoded	✓ Denial of Service (DoS)
✓ Divide before multiply	✓ Oracle Manipulation
✓ Integer overflow/underflow	✓ Signature Replay Attacks
✓ ERC's conformance	✓ Improper Handling of External Calls
✓ Missing Zero Address Validation	✓ Proxy Storage Collision
✓ Revert/require functions	✓ Use of Deprecated Functions

Disclaimer

This report is based on the code at the commit listed in the Scope section and on information the client provided during the review. It reflects Safe Edges' assessment at the time of the review and is not a guarantee that the code is free of vulnerabilities.

Changes made after the reviewed commit, deployment configuration, off-chain components and third-party dependencies are outside the scope of this report unless stated otherwise.

Security is ongoing. We recommend continued testing, monitoring and bug bounty coverage alongside any audit.

This report is not investment advice and is not an endorsement of the project, its team or any token.



Safe Edges is a Web3 security firm and official security partner for the Fuel, ENI and XI EcoChain ecosystems. We work across smart contract audits, Rust and WASM reviews, ZK circuit review, AI and MCP security, and penetration testing.

200+

PROJECTS AUDITED

\$4B+

TVL SECURED

0

POST-AUDIT HACKS