BINENET

Smart Contract Audit Report

Arbitrum L2

OpenOceanExchange

V 1.0

SC.3e4bdd381049

July 22nd, 2024

Table Of Contents

1 Report Overview

Binenet security team has audited the OpenOceanExchange, no risks was identified in OpenOceanExchange. users should pay attention to the following aspects when interacting with this project.

Contract Code	Function	Security Level	Status	Fix Result
OpenOceanExchangeProxy.sol	upgradeTo	Info	Audited	---
OpenOceanExchangeProxy.sol	upgradeToAndCall	Info	Audited	---
OpenOceanExchange.sol	rescueFunds	Info	Audited	---

***Risk Description:** Due to the use of proxy and logical architecture in the contract, the ProxyAdmin contract can achieve real-time upgrade of logical contracts, with a focus on the administrator's ability to handle permission changes.

2 Asset Management Security Assessment

Asset Type	Function	Security Level
User-Mortgaged Token Assets	OpenOceanExchange.rescueFunds	Info
Platform-Mortgaged Currency Assets	OpenOceanExchange.rescueFunds	Info

***Description:** Inspect the security measures for the management of digital currency assets within the contract business logic. Look for any security vulnerabilities that might lead to the loss of customer funds, such as improper recording of digital currency assets upon transfer into the contract or accidental transfer of assets out of the contract.

3 Audit Overview

3.1 Project Information

OpenOcean is the leading DEX aggregator, integrating 1000+ liquidity sources across 30+ blockchains into one seamless trading interface, to bring users the best swap returns on their DeFi trading.

This serves as a reference implementation of the OpenOceanExchange standard on the Arbitrum network.

3.2 Audit Information

Project Name	OpenOceanExchange
Platform	Arbitrum
Audit Scope	OpenOceanExchangeProxy https://arbiscan.io/address/0x6352a56caadc4f1e25cd6c75970fa768a3304e64#code OpenOceanExchange https://arbiscan.io/address/0xe21328bd90de1433f99512608558ff9481d94be2#code
Website	https://openocean.finance/

3.3 External Visibility Analysis

Function	Visibility	State Change	Modifier	Payable	Description
admin	external	false	ifAdmin	---	OpenOceanExchangeProxy
changeAdmin	external	true	ifAdmin	---	---
implementation	external	true	ifAdmin	---	---

upgradeTo	external	true	ifAdmin	---	---
upgradeToAndCall	external	true	ifAdmin	---	---
callUniswap	public	true	---	payable	OpenOceanExchange
callUniswapTo	public	true	---	payable	---
callUniswapToWithPermit	external	true	---	---	---
callUniswapWithPermit	external	true	---	---	---
initialize	public	true	initializer	---	---
pause	external	true	onlyOwner	---	---
renounceOwnership	public	true	onlyOwner	---	---
rescueFunds	external	true	onlyOwner	---	---
swap	external	true	whenNotPaused	payable	---
swapGmxV2	external	true	whenNotPaused	payable	---
transferOwnership	public	true	onlyOwner	---	---
uniswapV3Swap	external	true	---	payable	---
uniswapV3SwapCallback	external	true	---	---	---
uniswapV3SwapTo	public	true	---	payable	---
uniswapV3SwapToWithPermit	external	true	---	---	---

3.4 Audit Process

Audit time: From July 19th to July 22nd, 2024.

Audit methods: Static Analysis, Dynamic Testing, Typical Case Testing and Manual Review.

Audit team: Binenet Security Team.

4 Security Finding Details

4.1 ProxyAdmin Contract Management Risk

Severity Level : Info

Lines : OpenOceanExchangeProxy.sol # L493,L504

Description: According to the business logic, addresses with management privileges can upgrade proxy contract and redirect logical contract. After further auditing, the proxy contract points to the management contract ProxyAdmin [0xC979fEC5] controlled by a single signed EOA [0x9986EE0C] address, which poses a single point of leakage risk. It is recommended to use multi signature addresses to control the ProxyAdmin contract.

```
491     * NOTE: Only the admin can call this function. See {ProxyAdmin-upgrade}.
492     */
493    function upgradeTo(address newImplementation) external virtual ifAdmin {
494        _upgradeTo(newImplementation);
495    }
496
497    /**
498     * @dev Upgrade the implementation of the proxy, and then call a function from the new implementation as specified
499     * by `data`, which should be an encoded function call. This is useful to initialize new storage variables in the
500     * proxied contract.
501     *
502     * NOTE: Only the admin can call this function. See {ProxyAdmin-upgradeAndCall}.
503     */
504    function upgradeToAndCall(address newImplementation, bytes calldata data) external payable virtual ifAdmin {
505        _upgradeTo(newImplementation);
506        Address.functionDelegateCall(newImplementation, data);
507    }
```

The following is one of the upgrade logs for logical contract:

https://arbiscan.io/tx/0x28bb61516f010c745b208dfd468ea1cac683d4fca8ba4ffedc0f93d359677de3

#	Name	Type	Data
0	proxy	address	0x6352a56caadC4F1E25CD6c75970Fa768A3304e64
1	implementation	address	0xE21328bd90De1433F99512608558ff9481D94Be2

Recommendations: Considering the issue of single signature address leakage, it is recommended to use a multi signature approach to control the ProxyAdmin contract to ensure that logical contracts are not maliciously replaced.

Status : Audited.

Fix Result: Info, This issue depends on the business logic, such as the adoption of MPC management solutions for EOA addresses.

4.2 Rescue Funds

Severity Level : Info

Lines : OpenOceanExchange.sol # L3533

Description: Considering special cases such as users mistakenly transferring assets to a trading contract, the contract provides an emergency asset extraction feature, the logic of which depends on the business functionality.

```
3533    function rescueFunds(IERC20 token, uint256 amount) external onlyOwner {
3534        token.universalTransfer(payable(msg.sender), amount);
3535    }
3536
```

Recommendations: Judging based on business logic.

Status : Audited.

Fix Result: ---

4.3 Swap Core Logic

Severity Level : Safe

Lines : OpenOceanExchange.sol # L3468

Description: By auditing the core logic of the swap, parameter checks and swap related preparations were carried out at the entrance, such as transferring assets to the core swap contract caller. All core logic of the swap was handled by the caller, and the contract address was in a custom state. After the swap was completed, the parameters and returnAmount variables were checked again to ensure that the swap function proceeded as expected.

```
3468 >  function swap(…
3472    ) external payable whenNotPaused returns (uint256 returnAmount) {
3473        require(desc.minReturnAmount > 0, "Min return should not be 0");
3474        require(calls.length > 0, "Call data should exist");
3475
3476        uint256 flags = desc.flags;
3477        IERC20 srcToken = desc.srcToken;
3478        IERC20 dstToken = desc.dstToken;
3479
3480        require(msg.value == (srcToken.isETH() ? desc.amount : 0), "Invalid msg.value");
3481
3482 >      if (flags & _SHOULD_CLAIM != 0) {…
3485        }
3486
3487        address dstReceiver = (desc.dstReceiver == address(0)) ? msg.sender : desc.dstReceiver;
3488        uint256 initialSrcBalance = (flags & _PARTIAL_FILL != 0) ? srcToken.universalBalanceOf(msg.sender) : 0;
3489        uint256 initialDstBalance = dstToken.universalBalanceOf(dstReceiver);
3490
3491        caller.makeCalls{value: msg.value}(calls);
3492
```

Recommendations: ---

Status : Audited.

Fix Result: ---

4.4 SwapGmxV2 Core Logic

Severity Level : Safe

Lines : OpenOceanExchange.sol # L3468

Description: By auditing the core logic of swapGmxV2, this function is similar to the swap logic. It performs parameter checks and swap related preparations at the entrance, such as transferring assets to the core swap contract caller. All core logic of the swap is handled by the caller, and the contract address is in a custom state. After the swap is completed, the parameters and returnAmount variables are checked again to ensure that the swap function proceeds as expected.

```
3541    function swapGmxV2(
3542        IOpenOceanCaller caller,
3543        SwapDescription calldata desc,
3544        IOpenOceanCaller.CallDescription[] calldata calls
3545    ) external payable whenNotPaused returns (uint256 returnAmount) {
3546        require(calls.length > 0, "Call data should exist");
3547        require(msg.value > 0, "Invalid msg.value");
3548
3549        uint256 flags = desc.flags;
3550        IERC20 srcToken = desc.srcToken;
3551        IERC20 dstToken = desc.dstToken;
3552
3553 >      if (flags & _SHOULD_CLAIM != 0) {…
3556        }
3557
3558        address dstReceiver = (desc.dstReceiver == address(0)) ? msg.sender : desc.dstReceiver;
3559        uint256 initialSrcBalance = (flags & _PARTIAL_FILL != 0) ? srcToken.universalBalanceOf(msg.sender) : 0;
3560        uint256 initialDstBalance = dstToken.universalBalanceOf(dstReceiver);
3561
3562        caller.makeCalls{value: msg.value}(calls);
3563
```

Recommendations: ---

Status : Audited.

Fix Result: ---

4.5 callUniswapTo Core Logic

Severity Level : Safe

Lines : OpenOceanExchange.sol # L1631

Description: The callUniswapTo function is used for transaction processing of UniswapV2 related coin pairs. The function implements core functions such as parameter checking, verification, and swap internally.

```
1631    function callUniswapTo(
1632        IERC20 srcToken,
1633        uint256 amount,
1634        uint256 minReturn,
1635        bytes32[] calldata /* pools */,
1636        address payable recipient
1637    ) public payable returns (uint256 returnAmount) {
1638        assembly {
1639            // solhint-disable-line no-inline-assembly
1640 >          function reRevert() {…
1643            }
1644
1645 >          function revertWithReason(m, len) {…
1650            }
1651
1652 >          function swap(emptyPtr, swapAmount, pair, reversed, numerator, dst) -> ret {…
1684            }
1685
1686 >          function callSwap(emptyPtr, token, srcAmount, swapCaller, receiver, min) -> ret {…
1835            }
1836
1837            let emptyPtr := mload(0x40)
1838            mstore(0x40, add(emptyPtr, 0xc0))
1839            returnAmount := callSwap(emptyPtr, srcToken, amount, caller(), recipient, minReturn)
1840        }
1841    }
```

Recommendations: ---

Status : Audited.

Fix Result: ---

4.6 uniswapV3SwapTo Core Logic

Severity Level : Safe

Lines : OpenOceanExchange.sol # L3202

Description: The uniswapV3SwapTo function is used for transaction processing of UniswapV3 related currency pairs. The function implements core functions such as parameter checking, verification, and swap internally, and also adds a callback function called uniswapV3SwapCallback.

```
3202    function uniswapV3SwapTo(
3203        address payable recipient,
3204        uint256 amount,
3205        uint256 minReturn,
3206        uint256[] calldata pools
3207    ) public payable returns (uint256 returnAmount) {
3208        uint256 len = pools.length;
3209        address dstToken;
3210        require(len > 0, "UniswapV3: empty pools");
3211        uint256 lastIndex = len - 1;
3212        returnAmount = amount;
3213        bool wrapWeth = pools[0] & _WETH_WRAP_MASK > 0;
3214        bool unwrapWeth = pools[lastIndex] & _WETH_UNWRAP_MASK > 0;
3215 >      if (wrapWeth) {…
3218 >      } else {…
3220        }
3221 >      if (len > 1) {…
3227 >      } else {…
3229        }
3230
3231        require(returnAmount >= minReturn, "UniswapV3: min return");
3232
3233        assembly {
```

Recommendations: ---

Status : Audited.

Fix Result: ---

5 Audit Categories

Categories	Subitems
Business Security	Transfer token function
	Mint token and burn token vulnerability
	Contract logic function
	Mining pool deposit and withdrawal function
	Reasonableness of agreement amendment
	Functional design
	Dos caused by time
	Insecure oracles and their design
	Deployer private key leak hazard
General Vulnerability	Compiler version security
	Redundant code
	Use of safemath library
	Not recommended encoding
	Use require/assert mistakely
	Fallback function safety
	tx.origin authentication
	Owner permission control
	Gas consumption detection
	Call injection attack
	Low-level function safety
	Additional token vulnerabilities
	Access control
	Numeric overflow detection
	Arithmetic precision error
	Misuse of random number detection

	Unsafe external call
	Variable override
	Uninitialized storage pointer
	Return value call validation
	Transaction order dependent detection
	Timestamp dependent attack
	Denial of service attack detection
	Fake recharge vulnerability detection
	Reentrancy Attack Detection
	Replay attack detection
	Reordering attack detection

6 Explanation Of Vulnerability Rating

Vulnerability Rating	Rating Description
High Risk Vulnerability	Vulnerabilities that can directly lead to the loss of token contracts or user funds include: overflow, reentrancy, and false recharge. These issues may result in the token value being nullified, or cause the loss of tokens through fraudulent exchanges, or the loss of ETH or other tokens, etc. Vulnerabilities that can result in the loss of ownership of token contracts include: flaws in the access control of key functions and call injection that leads to the bypassing of access controls for key functions. Vulnerabilities that can cause token contracts to malfunction include: denial of service vulnerabilities caused by sending ETH to malicious addresses, and denial of service vulnerabilities due to gas exhaustion.
Medium Risk Vulnerability	Vulnerabilities that require specific addresses to trigger include scenarios such as overflow, which can only be initiated by the token contract owners. Additionally, there are access control flaws in non-critical functions and logical design flaws that do not directly lead to financial losses.
Low Risk Vulnerability	Vulnerabilities that are challenging to trigger include those that necessitate substantial amounts of ETH or tokens, such as overflow vulnerabilities. Additionally, there are vulnerabilities that, once triggered, do not directly benefit the attacker, such as overflow exploits

	from which the attacker cannot profit. Furthermore, there are transaction sequence-dependent risks, which are triggered by specifying a high gas wait.

7 Statement

Binenet issues this report solely based on the facts that have occurred or existed prior to the report's issuance and assumes corresponding responsibilities for them. We cannot assess the security status of the smart contract for any facts that occur or exist after the report is published, and we will not be held responsible for them.

This report does not cover external contract calls, new types of attacks that may emerge in the future, or contract upgrades and tampered codes (as the project evolves, smart contracts may introduce new pools, functional modules, external contract calls, etc.), nor does it include front-end or server security.

Binenet assumes that the documents and materials provided by the information provider as of the date of this report are complete and unaltered. If the provided information is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, Binenet shall not be liable for any losses or adverse effects arising from such discrepancies.

8 About Binenet

Founded in June 2023, Binenet is a dedicated and pure play blockchain security company. We focus on accurate, efficient, and intelligent blockchain threat detection and response. Committed to providing users with professional products and dedicated services in the field of blockchain security, our business functions include penetration testing, code auditing, emergency response, on-chain data monitoring, and AML (anti-money laundering), covering all aspects of blockchain ecosystem security.

Official Website
https://binenet.com

Telegram
https://t.me/binenetxyz

Twitter
https://twitter.com/binenetxyz

E-mail
team@binenet.com