OLYXAI

Smart Contract Audit Report

SYSFIXED

03 Feb 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
OLYXAI	OLYX	BSC

Addresses

Contract address	0x18606a5312870d2e0d1891868Fe6236713EdDD9C
Contract deployer address	0xA794933925856F85c75C8a81c378eb3F7F188Cd3

Project Website

https://olyx.ai/

Codebase

https://bscscan.com/address/0x18606a5312870d2e0d1891868Fe6236713EdDD9C#code

SUMMARY

Olyx is a revolutionary new cryptocurrency that combines the power of Artificial Intelligence(AI) with the speed and security of the Binance Smart Chain. But that's not all - the Olyx team is also offering a unique crowdfunding trade feature, allowing investors to contribute to a trading vault and reap the rewards of successful trades. The advantage is audited, 3% low buy tax, has no unlocked tokens, lp locked for 1 year, has no dev wallet, staking live, dapp ready, and multisig.

Contract Summary

Documentation Quality

OLYXAI provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also dont have any high risk issue.

Code Quality

The Overall quality of the basecode is standard.

- Standart solidity basecode and rules are already followed with OLYXAI with the discovery of several low issues.

Test Coverage

Test coverage of the project is 100% (Through Codebase)

Audit Findings Summary

- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 497, 501, 515, 534, 535, 546, 546, 564, 564, 654, 655, 656, 721, 742, 742, 743 and 754.
- SWC-110 | It is recommended to use use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 764 and 764.

CONCLUSION

We have audited the Olyx which has released on January 2023 to discover issues and identify potential security vulnerabilities in Olyx Project. This process is used to find bugs, technical issues, and security loopholes that find some common issues in the code.

The security audit report produced satisfactory results with a low risk issue on the contract project.

The most common issue found in writing code on contracts that do not pose a big risk, writing on contracts is close to the standard of writing contracts in general. Some of the low issues that were found were just arithmetic operations discovered. We recommended using recommended standard solidity arithmetic operation.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	PASS
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recursive call.	PASS
Assert Violation	SWC-110	Properly functioning code should never reach a failing assert statement.	ISSUE FOUND
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Caller	SWC-112	Delegatecalls should only be allowed to trusted addresses.	PASS
DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS

Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS

SMART CONTRACT ANALYSIS

Started	Sun Jan 8 2023 11:10:11GMT+0000 (Coordinated Universal Time)
Finished	Mon Jan 9 2023 12:11:10 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	OLYX.Sol

Detected Issues

ID	Title	Severity	Status
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 497

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
496
497   function decreaseAllowance(address spender, uint256 subtractedValue) public virtual
returns (bool) {
498   uint256 currentAllowance = _allowances[_msgSender()][spender];
499   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
500   unchecked {
501
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 501

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
500   unchecked {
501   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
502   }
503
504   return true;
505
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 515

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
514
515   _beforeTokenTransfer(sender, recipient, amount);
516
517   uint256 senderBalance = _balances[sender];
518   require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
519
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 534

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
533
534   _totalSupply += amount;
535   _balances[account] += amount;
536   emit Transfer(address(0), account, amount);
537
538
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 535

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
534   _totalSupply += amount;
535   _balances[account] += amount;
536   emit Transfer(address(0), account, amount);
537
538   _afterTokenTransfer(address(0), account, amount);
539
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 546

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
545
546   uint256 accountBalance = _balances[account];
547   require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
548   unchecked {
549   _balances[account] = accountBalance - amount;
550
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 564

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
563   require(owner != address(0), "ERC20: approve from the zero address");
564   require(spender != address(0), "ERC20: approve to the zero address");
565
566   _allowances[owner][spender] = amount;
567   emit Approval(owner, spender, amount);
568
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 564

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
563   require(owner != address(0), "ERC20: approve from the zero address");
564   require(spender != address(0), "ERC20: approve to the zero address");
565
566   _allowances[owner][spender] = amount;
567   emit Approval(owner, spender, amount);
568
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 654

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
653   require(token != address(this), "Owner cannot claim contract's balance of its own
tokens");
654   if (token == address(0x0)) {
655   payable(msg.sender).sendValue(address(this).balance);
656   return;
657   }
658
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 655

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
654   if (token == address(0x0)) {
655   payable(msg.sender).sendValue(address(this).balance);
656   return;
657   }
658   IERC20 ERC20token = IERC20(token);
659
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 656

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
655   payable(msg.sender).sendValue(address(this).balance);
656   return;
657   }
658   IERC20 ERC20token = IERC20(token);
659   uint256 balance = ERC20token.balanceOf(address(this));
660
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 721

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
720   } else if (from == uniswapV2Pair) {
721   _totalFees = marketingFeeOnBuy;
722   } else if (to == uniswapV2Pair) {
723   _totalFees = marketingFeeOnSell;
724   } else {
725
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 742

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
741
742   function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
743   require(newAmount > totalSupply() / 1_000_000, "SwapTokensAtAmount must be greater
than 0.0001% of total supply");
744   swapTokensAtAmount = newAmount;
745
746
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 743

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
742   function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
743   require(newAmount > totalSupply() / 1_000_000, "SwapTokensAtAmount must be greater
than 0.0001% of total supply");
744   swapTokensAtAmount = newAmount;
745
746   emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
747
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 754

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- OLYX.Sol

Locations

```
753
754   uniswapV2Router.swapExactTokensForTokens(
755   tokenAmount,
756   0,
757   path,
758
```

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to, or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without Sysfixed's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Sysfixed to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model, or legal compliance.

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Sysfixed and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Sysfixed) owe no duty of care.

ABOUT US

Sysfixed is a blockchain security certification organization established in 2021 with the objective to provide smart contract security services and verify their correctness in blockchain-based protocols. Sysfixed automatically scans for security vulnerabilities in Ethereum and other EVM-based blockchain smart contracts. Sysfixed a comprehensive range of analysis techniques—including static analysis, dynamic analysis, and symbolic execution—can accurately detect security vulnerabilities to provide an in-depth analysis report. With a vibrant ecosystem of world-class integration partners that amplify developer productivity, Sysfixed can be utilized in all phases of your project's lifecycle. Our team of security experts is dedicated to the research and improvement of our tools and techniques used to fortify your code.