YuanXiaoTu

Smart Contract Audit Report

SYSFIXED

12 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
YuanXiaoTu	YXT	Binance Smart Chain

Addresses

Contract address	0xaCDAc7BAbfABF21E9D0854d858824F98d352C7Ca
Contract deployer address	0x3e4Eef90fC1e1F050b6e03d4D6a6bc22EA10e4A8

Project Website

https://t.me/YSBP888

Codebase

https://bscscan.com/address/0xaCDAc7BAbfABF21E9D0854d858824F98d352C7Ca#code

SUMMARY

Pre-sale guaranteed compensation for the first project in the community, all projects in the community, white list pre-sale secured payment, do what you say!

Contract Summary

Documentation Quality

YuanXiaoTu 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.

- Standard solidity basecode and rules are already followed by YuanXiaoTu 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 572, 582, 590, 609, 611, 623, 624, 638, 640, 793, 809, 810, 877, 934, 941, 946, 951, 956, 983, 984, 994, 1001, 1002, 1019, 1050 and 1072.
- SWC-110 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 1007, 1008, 1039, 1040, 1041, 1061, 1062 and 1063.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 902, 967 and 973.

CONCLUSION

We have audited the YuanXiaoTu project released on January 2023 to discover issues and identify potential security vulnerabilities in YuanXiaoTu Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides a satisfactory result with some low-risk issues.

The issues found in the code on YuanXiaoTu smart contract do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, weak sources of randomness and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.

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 Callee	SWC-112	Delegate calls 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.	ISSUE FOUND
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	Wednesday Jan 11 2023 14:57:00 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Jan 12 2023 16:04:13 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	SAFU.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	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
571   unchecked {
572   _approve(sender, _msgSender(), currentAllowance - amount);
573   }
574   }
575
576
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 582

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
581   function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
582   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
583   return true;
584   }
585
586
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 590

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
589   unchecked {
590   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
591   }
592
593   return true;
594
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 609

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
608   unchecked {
609   _balances[sender] = senderBalance - amount;
610   }
611   _balances[recipient] += amount;
612
613
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 611

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
610   }
611   _balances[recipient] += amount;
612
613   emit Transfer(sender, recipient, amount);
614
615
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
622
623   _totalSupply += amount;
624   _balances[account] += amount;
625   emit Transfer(address(0), account, amount);
626
627
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 624

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
623   _totalSupply += amount;
624   _balances[account] += amount;
625   emit Transfer(address(0), account, amount);
626
627   _afterTokenTransfer(address(0), account, amount);
628
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 638

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
637   unchecked {
638   _balances[account] = accountBalance - amount;
639   }
640   _totalSupply -= amount;
641
642
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 640

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
639   }
640   _totalSupply -= amount;
641
642   emit Transfer(account, address(0), amount);
643
644
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 793

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
792
793   totalFeesOnBuySell    = liquidityFeeOnBuySell + marketingFeeOnBuySell +
communityFeeOnBuySell + burnFeeOnBuySell;
794
795   walletToWalletTransferFee = walletToWalletTransferFee_;
796
797
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 809

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
808
809   _mint(owner(), totalSupply_ * (10 ** decimals()));
810   swapTokensAtAmount = totalSupply() / 5_000;
811   }
812
813
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 810

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
809   _mint(owner(), totalSupply_ * (10 ** decimals()));
810   swapTokensAtAmount = totalSupply() / 5_000;
811   }
812
813   receive() external payable {
814
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 877

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
876
877   totalFeesOnBuySell = liquidityFeeOnBuySell + marketingFeeOnBuySell + communityFeeOnBuySell + burnFeeOnBuySell;
878
879   if (walletToWallet) {
880   require(walletToWalletTransferFee != 0, "walletToWalletTransferFee is zero");
881
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 934

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
933   totalFeesOnBuySell > 0 &&
934   totalFeesOnBuySell == (liquidityFeeOnBuySell + marketingFeeOnBuySell +
communityFeeOnBuySell + burnFeeOnBuySell)
935   ) {
936   swapping = true;
937
938
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 941

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
940   if (liquidityFeeOnBuySell > 0) {
941   uint256 liquidityTokens = contractTokenBalance * liquidityFeeOnBuySell / totalFeesOnBuySell;
942   swapAndLiquify(liquidityTokens);
943   }
944
945
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 946

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
945   if (marketingFeeOnBuySell > 0) {
946   uint256 marketingTokens = contractTokenBalance * marketingFeeOnBuySell / totalFeesOnBuySell;
947   swapAndSendMarketing(marketingTokens);
948   }
949
950
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 951

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
950   if (communityFeeOnBuySell > 0) {
951   uint256 communityTokens = contractTokenBalance * communityFeeOnBuySell /
totalFeesOnBuySell;
952   swapAndSendCommunity(communityTokens);
953   }
954
955
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 956

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
955   if (burnFeeOnBuySell > 0) {
956   uint256 burnTokens = contractTokenBalance * burnFeeOnBuySell / totalFeesOnBuySell;
957   super._transfer(address(this), address(0xdead), burnTokens);
958   }
959
960
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 983

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
982   if (_totalFees > 0) {
983   uint256 fees = (amount * _totalFees) / 100;
984   amount = amount - fees;
985   super._transfer(from, address(this), fees);
986   }
987
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 984

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
983   uint256 fees = (amount * _totalFees) / 100;
984   amount = amount - fees;
985   super._transfer(from, address(this), fees);
986   }
987
988
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 994

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
993   function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
994   require(newAmount > totalSupply() / 1_000_000, "SwapTokensAtAmount must be greater than 0.0001% of total supply");
995   swapTokensAtAmount = newAmount;
996
997   emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
998
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1001

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
1000   function swapAndLiquify(uint256 tokens) private {
1001   uint256 half = tokens / 2;
1002   uint256 otherHalf = tokens - half;
1003
1004   uint256 initialBalance = erc20UsdtToken.balanceOf(address(this));
1005
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1002

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
1001   uint256 half = tokens / 2;
1002   uint256 otherHalf = tokens - half;
1003
1004   uint256 initialBalance = erc20UsdtToken.balanceOf(address(this));
1005
1006
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1019

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
1018   tRecv.getToken(erc20UsdtToken);
1019   uint256 newBalance = erc20UsdtToken.balanceOf(address(this)) - initialBalance;
1020
1021   uniswapV2Router.addLiquidity(
1022   address(this),
1023
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1050

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
1049
1050   uint256 newBalance = address(this).balance - initialBalance;
1051
1052   payable(marketingWallet).sendValue(newBalance);
1053
1054
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1072

low SEVERITY

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

Source File

- SAFU.sol

Locations

```
1071
1072   uint256 newBalance = address(this).balance - initialBalance;
1073
1074   payable(communityWallet).sendValue(newBalance);
1075
1076
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1007

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1006   address[] memory path = new address[](2);
1007   path[0] = address(this);
1008   path[1] = address(erc20UsdtToken);
1009
1010   uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
1011
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1008

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1007   path[0] = address(this);
1008   path[1] = address(erc20UsdtToken);
1009
1010   uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
1011   half,
1012
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1039

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1038   address[] memory path = new address[](3);
1039   path[0] = address(this);
1040   path[1] = address(erc20UsdtToken);
1041   path[2] = uniswapV2Router.WETH();
1042
1043
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1040

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1039   path[0] = address(this);
1040   path[1] = address(erc20UsdtToken);
1041   path[2] = uniswapV2Router.WETH();
1042
1043   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1044
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1041

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1040   path[1] = address(erc20UsdtToken);
1041   path[2] = uniswapV2Router.WETH();
1042
1043   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1044   tokenAmount,
1045
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1061

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1060   address[] memory path = new address[](3);
1061   path[0] = address(this);
1062   path[1] = address(erc20UsdtToken);
1063   path[2] = uniswapV2Router.WETH();
1064
1065
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1062

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1061   path[0] = address(this);
1062   path[1] = address(erc20UsdtToken);
1063   path[2] = uniswapV2Router.WETH();
1064
1065   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1066
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1063

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- SAFU.sol

Locations

```
1062   path[1] = address(erc20UsdtToken);
1063   path[2] = uniswapV2Router.WETH();
1064
1065   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1066   tokenAmount,
1067
```

SWC-120 | POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.

LINE 902

low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source File

- SAFU.sol

Locations

```
901   require(makeOfferBlock == 0, "Have started trading");
902   makeOfferBlock = block.number;
903
904   emit MakeOffer(makeOfferBlock);
905   }
906
```

SWC-120 | POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.

LINE 967

low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source File

- SAFU.sol

Locations

```
966   } else if (from == uniswapV2Pair) {
967   if (makeOfferBlock == 0 || block.number < makeOfferBlock) {
968   _totalFees = 89;
969   } else {
970   _totalFees = totalFeesOnBuySell;
971
```

SWC-120 | POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.

LINE 973

low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source File

- SAFU.sol

Locations

```
972   } else if (to == uniswapV2Pair) {
973   if (makeOfferBlock == 0 || block.number < makeOfferBlock) {
974   _totalFees = 89;
975   } else {
976   _totalFees = totalFeesOnBuySell;
977
```

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.