Child Support

Smart Contract Audit Report

SYSFIXED

21 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Child Support	CS	BSC

Addresses

Contract address	0x502B8136c48977b975a6C62B08Ac4E15Dabc8172
Contract deployer address	0xacAF2183e1513f4AB4258500BbD6E9E6aac0a213

Project Website

https://child.support/

Codebase

https://bscscan.com/address/0x502B8136c48977b975a6C62B08Ac4E15Dabc8172#code

SUMMARY

Child Support is a platform that connects the non-profit community, full transparency & traceability of donations. It's a blockchain project focused on raising money for charities for children and aiming to create a moderate platform for both trading of NFT and a transparent charity platform. Child Support is a real community Exchange platform, with NFT collection and with other features like redistribution reward and burn.

Contract Summary

Documentation Quality

This project has a sufficient amount of documentation.

- Technical description provided.

Code Quality

The overall quality of the code in this project is up to standard.

- The official Solidity style guide is followed.

Test Scope

Project test coverage is 100% (Via Codebase).

Audit Findings Summary

Issues Found

- SWC-101 | Arithmetic operation issues discovered on lines 577, 579, 724, and 871.
- SWC-108 | State variable visibility is not set on line 574. It is best practice to set the visibility of state variables explicitly to public or private.
- SWC-110 | Out of bounds array access discovered on line 804.

CONCLUSION

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

The security audit report produced satisfactory results with low-risk issues.

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. The low-level issue found is out of bounds array access which the index access expression can cause an exception in case of 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.	ISSUE FOUND
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
Check-Effect Interaction	SWC-107	Check-Effect-Interaction pattern should be followed if the code performs ANY external 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	Fri Jan 20 2023 20:37:20 GMT+0000 (Coordinated Universal Time)
Finished	Sat Jan 21 2023 00:02:50 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	child.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-108	STATE VARIABLE VISIBILITY IS NOT SET	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 577

low SEVERITY

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

Source File

- child.sol

Locations

```
576   bool public swapAndLiquifyEnabled = true;
577   uint256 public noOfTokensSellToGetReward = 1_000_000 * 10**_decimals;
578   event MinTokensBeforeSwapUpdated(uint256 minTokensBeforeSwap);
579   |
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 579

low SEVERITY

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

Source File

- child.sol

Locations

```
578   uint256 public noOfTokensSellToGetReward = 1_000_000 * 10**_decimals;
579   event MinTokensBeforeSwapUpdated(uint256 minTokensBeforeSwap);
580   event SwapAndLiquifyEnabledUpdated(bool enabled);
581   event SwapAndLiquify(
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 724

low SEVERITY

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

Source File

- child.sol

Locations

```
723   "Excluded addresses cannot call this function"
724   );
725   (uint256 rAmount, , , , ) = _getValues(tAmount);
726   _rOwned[sender] = _rOwned[sender] - (rAmount);
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 871

low SEVERITY

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

Source File

- child.sol

Locations

```
870   rSupply = rSupply - (_rOwned[_excluded[i]]);
871   tSupply = tSupply - (_tOwned[_excluded[i]]);
872   }
873   if (rSupply < _rTotal / (_tTotal)) return (_rTotal, _tTotal);
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET

LINE 574

low SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.

Source File

- child.sol

Locations

```
573   address public uniswapV2Pair;
574   bool inSwapAndLiquify;
575   bool public swapAndLiquifyEnabled = true;
576   |
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 804

low SEVERITY

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

Source File

- child.sol

Locations

```
803   function _reflectFee(uint256 rFee, uint256 tFee) private {
804   _rTotal = _rTotal - (rFee);
805   _tFeeTotal = _tFeeTotal + (tFee);
806   }
```

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.