



Socaverse

Smart Contract Audit Report

TABLE OF CONTENTS

[Audited Details](#)

- Audited Project
- Blockchain
- Addresses
- Project Website
- Codebase

[Summary](#)

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

[Conclusion](#)

[Audit Results](#)

[Smart Contract Analysis](#)

- Detected Vulnerabilities

[Disclaimer](#)

[About Us](#)

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Socaverse	SOCA	Binance Smart Chain

Addresses

Contract address	0x6598463d6cbe4b51e9977437bf1200df4c45286c
Contract deployer address	0x21BbEbD8A251c1b7F0e4464bfe160Ca57077f818

Project Website

<https://www.socaverse.one/>

Codebase

<https://bscscan.com/address/0x6598463d6cbe4b51e9977437bf1200df4c45286c#code>

SUMMARY

Metaverse is a tenacious, infinitely-scaling virtual space designed with its economy and identity system. This concept has been booming since 2020 due to the development of Web 3.0 technology, NFTs, Play-to-Earn games, and infrastructures. Projects like Decentraland, The Sandbox, and Roblox remarkably demonstrate the potential of the metaverse. Currently, the market cap of the metaverse is \$30.7 billion, which is expected to reach \$280 billion in 2025 with the development of new technology and the growing demand for the connection between the natural world and the crypto world. It is believed that we will always be connected to the metaverse, extending our senses of sight, sound, and touch, blending digital items into the physical world, or popping into fully immersive 3D environments whenever we want. Despite its fantastic future, the adoption of the metaverse is still relatively low. The adoption of a technology depends on the capability and incentive. For power, companies such as Facebook, Nvidia, Decentraland, Metamask have been devoted to building infrastructures for it. For motivation, Play-to-Earn games like Axie Infinity, NFT collectibles like NBA top shot, and NFT avatar identities like CryptoPunks have been exploring new ways to attract more non-crypto users to the Web 3.0 world. To speed up the development of the metaverse, we are leveraging it to establish our platform. Socaverse is the first blockchain-driven metaverse for soccer lovers. We aim to build a vertical metaverse that attracts non-crypto soccer lovers to play and earn through multiple revenue streams.

Contract Summary

Documentation Quality

Socaverse 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 Socaverse 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 140, 140, 141, 141, 151, 151, 153, 153, 154, 154, 256, 262, 269, 292, 306, 308, 339, 340, 344, 348,

350, 354, 358, 360, 376, 376, 377, 377, 378, 378, 379, 379, 379, 384, 390, 391, 392, 393, 393, 399, 405, 407, 408, 410, 433, 438, 450, 453, 456, 457, 468, 474, 474, 475, 475, 476, 482, 483, 483, 484, 491, 519, 526, 536, 540, 540, 544, 544, 557, 603 and 308.



- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 7.
- SWC-110 SWC-123 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 307, 308, 308, 406, 406, 407, 408, 516, 517 and 558.

CONCLUSION

We have audited the Socaverse Project released on April 2022 to discover issues and identify potential security vulnerabilities in Socaverse 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 satisfactory results with low-risk issues.

The issues found in the Socaverse smart contract code 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, a floating pragma is set, a state variable visibility is not set, 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. The current pragma Solidity directive is `^0.8.10`. Specifying a fixed compiler version is recommended to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

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.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
Unprotected Ether Withdrawal	SWC-105	Due to missing or insufficient access controls, malicious parties can withdraw from the contract.	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
Uninitialized Storage Pointer	SWC-109	Uninitialized local storage variables can point to unexpected storage locations in the contract.	PASS
Assert Violation	SWC-110 SWC-123	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	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
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	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
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations.	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
Insufficient Gas Griefing	SWC-126	Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract.	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.	PASS

Typographical Error	SWC-129	A typographical error can occur for example when the intent of a defined operation is to sum a number to a variable.	PASS
Override control character	SWC-130	Malicious actors can use the Right-To-Left-Override unicode character to force RTL text rendering and confuse users as to the real intent of a contract.	PASS
Unused variables	SWC-131 SWC-135	Unused variables are allowed in Solidity and they do not pose a direct security issue.	PASS
Unexpected Ether balance	SWC-132	Contracts can behave erroneously when they strictly assume a specific Ether balance.	PASS
Hash Collisions Variable	SWC-133	Using <code>abi.encodePacked()</code> with multiple variable length arguments can, in certain situations, lead to a hash collision.	PASS
Hardcoded gas amount	SWC-134	The <code>transfer()</code> and <code>send()</code> functions forward a fixed amount of 2300 gas.	PASS
Unencrypted Private Data	SWC-136	It is a common misconception that private type variables cannot be read.	PASS

SMART CONTRACT ANALYSIS

Started	Tuesday Apr 26 2022 22:32:45 GMT+0000 (Coordinated Universal Time)
Finished	Wednesday Apr 27 2022 20:34:12 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Socaverse.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-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-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 140

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
139
140  uint256 private _tTotal = 1e13 * 10 ** _decimals;
141  uint256 private _rTotal = (MAX - (MAX % _tTotal));
142
143  IERC20 public tokenV1 = IERC20(0x57Af121A8dDb1F9A8Fafcf3229c92bF2856A8a29);
//address of the old version
144
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 140

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
139
140  uint256 private _tTotal = 1e13 * 10 ** _decimals;
141  uint256 private _rTotal = (MAX - (MAX % _tTotal));
142
143  IERC20 public tokenV1 = IERC20(0x57Af121A8dDb1F9A8Fafcf3229c92bF2856A8a29);
//address of the old version
144
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 141

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
140 uint256 private _tTotal = 1e13 * 10 ** _decimals;
141 uint256 private _rTotal = (MAX - (MAX % _tTotal));
142
143 IERC20 public tokenV1 = IERC20(0x57Af121A8dDb1F9A8Fafcf3229c92bF2856A8a29);
//address of the old version
144 uint256 public amountV1Migrated; // amount of tokens V1 migrated
145
```


SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 141

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
140 uint256 private _tTotal = 1e13 * 10 ** _decimals;
141 uint256 private _rTotal = (MAX - (MAX % _tTotal));
142
143 IERC20 public tokenV1 = IERC20(0x57Af121A8dDb1F9A8Fafcf3229c92bF2856A8a29);
//address of the old version
144 uint256 public amountV1Migrated; // amount of tokens V1 migrated
145
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 151

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
150
151  uint256 public swapTokensAtAmount = 20_000_000 * 10 ** _decimals;
152
153  uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154  uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 151

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
150
151  uint256 public swapTokensAtAmount = 20_000_000 * 10 ** _decimals;
152
153  uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154  uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 153

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
152
153  uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154  uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
156  address public marketingAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
157
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 153

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
152
153  uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154  uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
156  address public marketingAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
157
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 154

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
153 uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154 uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
156 address public marketingAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
157 address public devAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
158
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 154

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
153 uint256 public maxBuyAmount = 100_000_000 * 10 ** _decimals;
154 uint256 public maxWalletBalance = 3000_000_000 * 10 ** _decimals;
155
156 address public marketingAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
157 address public devAddress = 0x0C0886A8dB6C57a4a97057517482856bb09ABF35;
158
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 256

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
255     require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
256     _approve(sender, _msgSender(), currentAllowance - amount);
257
258     return true;
259 }
260
```


SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 262

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
261  function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
262  _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
263  return true;
264  }
265
266
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 269

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
268     require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
269     _approve(_msgSender(), spender, currentAllowance - subtractedValue);
270
271     return true;
272 }
273
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 292

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
291     uint256 currentRate = _getRate();
292     return rAmount / currentRate;
293 }
294
295     function excludeFromReward(address account) public onlyOwner() {
296
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 306

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
305   require(!_isExcluded[account], "Account is not excluded");
306   for (uint256 i = 0; i < _excluded.length; i++) {
307     if (_excluded[i] == account) {
308       _excluded[i] = _excluded[_excluded.length - 1];
309       _tOwned[account] = 0;
310
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 308

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
307   if (_excluded[i] == account) {  
308     _excluded[i] = _excluded[_excluded.length - 1];  
309     _tOwned[account] = 0;  
310     _isExcluded[account] = false;  
311     _excluded.pop();  
312
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 339

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
338 function _reflectRfi(uint256 rRfi, uint256 tRfi) private {
339     _rTotal -= rRfi;
340     totFeesPaid.rfi += tRfi;
341 }
342
343
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 340

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
339  _rTotal -= rRfi;  
340  totFeesPaid.rfi += tRfi;  
341  }  
342  
343  function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {  
344
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 344

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
343     function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {
344         totFeesPaid.liquidity += tLiquidity;
345
346         if (!_isExcluded[address(this)])
347         {
348
```


SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 348

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
347 {  
348   _tOwned[address(this)] += tLiquidity;  
349 }  
350   _rOwned[address(this)] += rLiquidity;  
351 }  
352
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 350

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
349     }
350     _rOwned[address(this)] += rLiquidity;
351     }
352
353     function _takeMarketing(uint256 rMarketing, uint256 tMarketing) private {
354
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
353     function _takeMarketing(uint256 rMarketing, uint256 tMarketing) private {
354         totFeesPaid.marketing += tMarketing;
355
356         if (!_isExcluded[address(this)])
357         {
358
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 358

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
357 {  
358   _tOwned[address(this)] += tMarketing;  
359 }  
360   _rOwned[address(this)] += rMarketing;  
361 }  
362
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 360

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
359     }
360     _rOwned[address(this)] += rMarketing;
361     }
362
363     function _getValues(uint256 tAmount, bool takeFee) private view returns
(valuesFromGetValues memory to_return) {
364
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 376

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
375
376 s.tRfi = tAmount * taxes.rfi / 100;
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 376

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
375
376 s.tRfi = tAmount * taxes.rfi / 100;
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 377

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
376 s.tRfi = tAmount * taxes.rfi / 100;
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 377

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
376 s.tRfi = tAmount * taxes.rfi / 100;
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 378

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381 }
382
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 378

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
377 s.tMarketing = tAmount * taxes.marketing / 100;
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381 }
382
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 379

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381 }
382
383
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 379

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381 }
382
383
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 379

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
378 s.tLiquidity = tAmount * taxes.liquidity / 100;
379 s.tTransferAmount = tAmount - s.tRfi - s.tMarketing - s.tLiquidity;
380 return s;
381 }
382
383
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 384

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
383 function _getRValues(valuesFromGetValues memory s, uint256 tAmount, bool takeFee,
uint256 currentRate) private pure returns (uint256 rAmount, uint256 rTransferAmount,
uint256 rRfi, uint256 rMarketing, uint256 rLiquidity) {
384     rAmount = tAmount * currentRate;
385
386     if (!takeFee) {
387         return (rAmount, rAmount, 0, 0, 0);
388     }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 390

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
389
390   rRfi = s.tRfi * currentRate;
391   rMarketing = s.tMarketing * currentRate;
392   rLiquidity = s.tLiquidity * currentRate;
393   rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;
394
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 391

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
390  rRfi = s.tRfi * currentRate;  
391  rMarketing = s.tMarketing * currentRate;  
392  rLiquidity = s.tLiquidity * currentRate;  
393  rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;  
394  return (rAmount, rTransferAmount, rRfi, rMarketing, rLiquidity);  
395
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 392

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
391  rMarketing = s.tMarketing * currentRate;
392  rLiquidity = s.tLiquidity * currentRate;
393  rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;
394  return (rAmount, rTransferAmount, rRfi, rMarketing, rLiquidity);
395  }
396
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 393

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
392  rLiquidity = s.tLiquidity * currentRate;
393  rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;
394  return (rAmount, rTransferAmount, rRfi, rMarketing, rLiquidity);
395  }
396
397
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 393

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
392  rLiquidity = s.tLiquidity * currentRate;
393  rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;
394  return (rAmount, rTransferAmount, rRfi, rMarketing, rLiquidity);
395  }
396
397
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 393

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
392  rLiquidity = s.tLiquidity * currentRate;
393  rTransferAmount = rAmount - rRfi - rMarketing - rLiquidity;
394  return (rAmount, rTransferAmount, rRfi, rMarketing, rLiquidity);
395  }
396
397
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 399

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
398 (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
399 return rSupply / tSupply;
400 }
401
402 function _getCurrentSupply() private view returns (uint256, uint256) {
403
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 405

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
404  uint256 tSupply = _tTotal;
405  for (uint256 i = 0; i < _excluded.length; i++) {
406  if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
407  rSupply = rSupply - _rOwned[_excluded[i]];
408  tSupply = tSupply - _tOwned[_excluded[i]];
409
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
406  if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
    (_rTotal, _tTotal);
407  rSupply = rSupply - _rOwned[_excluded[i]];
408  tSupply = tSupply - _tOwned[_excluded[i]];
409  }
410  if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
411
```


SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 408

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
407     rSupply = rSupply - _rOwned[_excluded[i]];
408     tSupply = tSupply - _tOwned[_excluded[i]];
409     }
410     if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
411     return (rSupply, tSupply);
412
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 410

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
409     }  
410     if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);  
411     return (rSupply, tSupply);  
412     }  
413  
414
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 433

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
432  if (to != pair) {  
433  require(balanceOf(to) + amount <= maxWalletBalance, "You are exceeding  
maxWalletBalance");  
434  }  
435  }  
436  
437
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
437     bool canSwap = balanceOf(address(this)) >= swapTokensAtAmount;
438     if (!swapping && swapEnabled && canSwap && from != pair &&
!_isExcludedFromFee[from] && !_isExcludedFromFee[to] && (taxes.liquidity +
taxes.marketing) > 0) {
439         swapAndLiquify(swapTokensAtAmount);
440     }
441
442
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 450

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
449     if (!_isExcluded[sender]) {
450         _tOwned[sender] = _tOwned[sender] - tAmount;
451     }
452     if (!_isExcluded[recipient]) {
453         _tOwned[recipient] = _tOwned[recipient] + s.tTransferAmount;
454     }
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 453

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
452     if (!_isExcluded[recipient]) {
453         _tOwned[recipient] = _tOwned[recipient] + s.tTransferAmount;
454     }
455
456     _rOwned[sender] = _rOwned[sender] - s.rAmount;
457
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 456

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
455
456  _rOwned[sender] = _rOwned[sender] - s.rAmount;
457  _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
458
459  if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
460
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 457

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
456  _rOwned[sender] = _rOwned[sender] - s.rAmount;
457  _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
458
459  if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
460  if (s.rLiquidity > 0 || s.tLiquidity > 0) {
461
```


SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 468

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
467     emit Transfer(sender, recipient, s.tTransferAmount);
468     emit Transfer(sender, address(this), s.tLiquidity + s.tMarketing);
469
470   }
471
472
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 474

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
473
474     uint256 denominator = (taxes.liquidity + taxes.marketing) * 2;
475     uint256 tokensToAddLiquidityWith = tokens * taxes.liquidity / denominator;
476     uint256 toSwap = tokens - tokensToAddLiquidityWith;
477
478
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 474

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
473
474     uint256 denominator = (taxes.liquidity + taxes.marketing) * 2;
475     uint256 tokensToAddLiquidityWith = tokens * taxes.liquidity / denominator;
476     uint256 toSwap = tokens - tokensToAddLiquidityWith;
477
478
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
474 uint256 denominator = (taxes.liquidity + taxes.marketing) * 2;  
475 uint256 tokensToAddLiquidityWith = tokens * taxes.liquidity / denominator;  
476 uint256 toSwap = tokens - tokensToAddLiquidityWith;  
477  
478 uint256 initialBalance = address(this).balance;  
479
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
474 uint256 denominator = (taxes.liquidity + taxes.marketing) * 2;
475 uint256 tokensToAddLiquidityWith = tokens * taxes.liquidity / denominator;
476 uint256 toSwap = tokens - tokensToAddLiquidityWith;
477
478 uint256 initialBalance = address(this).balance;
479
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 476

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
475 uint256 tokensToAddLiquidityWith = tokens * taxes.liquidity / denominator;  
476 uint256 toSwap = tokens - tokensToAddLiquidityWith;  
477  
478 uint256 initialBalance = address(this).balance;  
479  
480
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 482

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
481
482  uint256 deltaBalance = address(this).balance - initialBalance;
483  uint256 unitBalance = deltaBalance / (denominator - taxes.liquidity);
484  uint256 bnbToAddLiquidityWith = unitBalance * taxes.liquidity;
485
486
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 483

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
482 uint256 deltaBalance = address(this).balance - initialBalance;
483 uint256 unitBalance = deltaBalance / (denominator - taxes.liquidity);
484 uint256 bnbToAddLiquidityWith = unitBalance * taxes.liquidity;
485
486 if (bnbToAddLiquidityWith > 0) {
487
```


SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 483

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
482 uint256 deltaBalance = address(this).balance - initialBalance;
483 uint256 unitBalance = deltaBalance / (denominator - taxes.liquidity);
484 uint256 bnbToAddLiquidityWith = unitBalance * taxes.liquidity;
485
486 if (bnbToAddLiquidityWith > 0) {
487
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 484

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
483     uint256 unitBalance = deltaBalance / (denominator - taxes.liquidity);
484     uint256 bnbToAddLiquidityWith = unitBalance * taxes.liquidity;
485
486     if (bnbToAddLiquidityWith > 0) {
487
488
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 491

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
490
491 uint256 marketingAmt = unitBalance * 2 * taxes.marketing;
492 if (marketingAmt > 0) {
493 payable(marketingAddress).transfer(marketingAmt);
494 }
495
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 491

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
490
491 uint256 marketingAmt = unitBalance * 2 * taxes.marketing;
492 if (marketingAmt > 0) {
493 payable(marketingAddress).transfer(marketingAmt);
494 }
495
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 536

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
535     function updateMaxWalletBalance(uint256 amount) external onlyOwner {
536         maxWalletBalance = amount * 10 ** _decimals;
537     }
538
539     function updatMaxBuyAmt(uint256 amount) external onlyOwner {
540
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 536

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
535     function updateMaxWalletBalance(uint256 amount) external onlyOwner {
536         maxWalletBalance = amount * 10 ** _decimals;
537     }
538
539     function updatMaxBuyAmt(uint256 amount) external onlyOwner {
540
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 540

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
539     function updatMaxBuyAmt(uint256 amount) external onlyOwner {
540         maxBuyAmount = amount * 10 ** _decimals;
541     }
542
543     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
544
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 540

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
539 function updatMaxBuyAmt(uint256 amount) external onlyOwner {
540     maxBuyAmount = amount * 10 ** _decimals;
541 }
542
543 function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
544
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 544

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
543     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
544         swapTokensAtAmount = amount * 10 ** _decimals;
545     }
546
547     function updateSwapEnabled(bool _enabled) external onlyOwner {
548
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 544

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
543     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
544         swapTokensAtAmount = amount * 10 ** _decimals;
545     }
546
547     function updateSwapEnabled(bool _enabled) external onlyOwner {
548
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 557

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
556 function bulkAntiBot(address[] memory accounts, bool state) external onlyOwner {
557     for (uint256 i = 0; i < accounts.length; i++) {
558         _isBot[accounts[i]] = state;
559     }
560 }
561
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 603

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
602  _tokenTransfer(address(this), msg.sender, amount, false);
603  amountV1Migrated += amount;
604  emit Migrated(msg.sender, amount);
605  }
606
607
```

SWC-101 | COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED

LINE 308

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
307   if (_excluded[i] == account) {  
308     _excluded[i] = _excluded[_excluded.length - 1];  
309     _tOwned[account] = 0;  
310     _isExcluded[account] = false;  
311     _excluded.pop();  
312
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 7

low SEVERITY

The current pragma Solidity directive is `""^0.8.10""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source File

- Socaverse.sol

Locations

```
6
7  pragma solidity ^0.8.10;
8
9  interface IERC20 {
10     function totalSupply() external view returns (uint256);
11
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 307

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
306   for (uint256 i = 0; i < _excluded.length; i++) {
307     if (_excluded[i] == account) {
308       _excluded[i] = _excluded[_excluded.length - 1];
309       _tOwned[account] = 0;
310       _isExcluded[account] = false;
311     }
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 308

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
307   if (_excluded[i] == account) {  
308     _excluded[i] = _excluded[_excluded.length - 1];  
309     _tOwned[account] = 0;  
310     _isExcluded[account] = false;  
311     _excluded.pop();  
312
```


SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 308

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
307   if (_excluded[i] == account) {  
308     _excluded[i] = _excluded[_excluded.length - 1];  
309     _tOwned[account] = 0;  
310     _isExcluded[account] = false;  
311     _excluded.pop();  
312
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 406

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
405   for (uint256 i = 0; i < _excluded.length; i++) {
406     if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
        (_rTotal, _tTotal);
407     rSupply = rSupply - _rOwned[_excluded[i]];
408     tSupply = tSupply - _tOwned[_excluded[i]];
409   }
410
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 406

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
405   for (uint256 i = 0; i < _excluded.length; i++) {
406     if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
        (_rTotal, _tTotal);
407     rSupply = rSupply - _rOwned[_excluded[i]];
408     tSupply = tSupply - _tOwned[_excluded[i]];
409   }
410
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 407

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
406  if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
    (_rTotal, _tTotal);
407  rSupply = rSupply - _rOwned[_excluded[i]];
408  tSupply = tSupply - _tOwned[_excluded[i]];
409  }
410  if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
411
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 408

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
407     rSupply = rSupply - _rOwned[_excluded[i]];
408     tSupply = tSupply - _tOwned[_excluded[i]];
409 }
410 if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
411 return (rSupply, tSupply);
412
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 516

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
515     address[] memory path = new address[](2);
516     path[0] = address(this);
517     path[1] = router.WETH();
518
519     _approve(address(this), address(router), tokenAmount);
520
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 517

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
516 path[0] = address(this);  
517 path[1] = router.WETH();  
518  
519 _approve(address(this), address(router), tokenAmount);  
520  
521
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 558

low SEVERITY

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

Source File

- Socaverse.sol

Locations

```
557   for (uint256 i = 0; i < accounts.length; i++) {  
558     _isBot[accounts[i]] = state;  
559   }  
560 }  
561  
562
```


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.