FROG CEO

Smart Contract Audit Report

SYSFIXED

09 Mar 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
FROG CEO	FROG CEO	Binance Smart Chain

Addresses

Contract address	0xbbf8b05ef7af53ccbff8e3673e73714f939bfd84
Contract deployer address	0x092Bd1A7624CA27291B08d4F7F2a1A597b9A9e1B

Project Website

https://www.gamiworld.io/

Codebase

https://bscscan.com/address/0xbbf8b05ef7af53ccbff8e3673e73714f939bfd84#code

SUMMARY

Frog Ceo, including but not limited to the overall project, token, Website, smart contracts, and any apps ("FROGCEO") as presented in this conceptual paper, is not a licensed, unlicensed, or exempted financial or payment service of any kind and in any jurisdiction. Any terminology used in this Whitepaper, on thWebsitete, or within the app is intended only as a primary reference, without any e active or legal meaning of the same terms in a regulated and economic environment. Frog Ceo is a community-driven project and does not have owners, shareholders, promoters, marketers, managers, directors, or other figures or entities exerting any form of governance. The Frog Ceo smart contracts are open-source, security audited, permanent, and non-modifiable in any way. The Frog Ceo token is strictly a utility token in all jurisdictions. It is not and cannot be considered a "security" or otherwise regulated token of any kind. Frog Ceo is not e-money and fiat money or an asset-backed stablecoin, whether global or limited in scope. This Whitepaper taken by itself is not a contract or a contractual agreement of any kind, nor is it an invitation, solicitation, or offer to invest in Frog Ceo or acquire or use Frog Ceo tokens in any way with any expectation of profit from that place.

Contract Summary

Documentation Quality

FROG CEO 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 FROG CEO 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 156, 156, 157, 158, 159, 159, 162, 162, 272, 277, 288, 318, 337, 338, 361, 361, 368, 370, 372, 400, 400, 402, 403, 407, 407, 425, 429, 429, 433, 433, 437, 442, 449, 451, 452, 506, 509, 510, 513, 554, 565, 565 and 338.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 28.

- 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 337, 338, 338, 443, 446, 450, 451, 537, 539 and 554.

CONCLUSION

We have audited the FROG CEO project released on March 2023 to discover issues and identify potential security vulnerabilities in FROG CEO 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 FROG CEO 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 a floating pragma 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.

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 abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision.	PASS
Hardcoded gas amount	SWC-134	The transfer() and send() 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	Wednesday Mar 08 2023 17:45:34 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Mar 09 2023 22:55:04 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	FROGCEO.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	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 156

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
155   address public deadWallet = 0x000000000000000000000000000000000000dEaD;
156   address public marketingWallet = 0xD8Ad22Ae19340ff77373bE33970aE92883Aba80A;
157
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 156

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
155   address public deadWallet = 0x000000000000000000000000000000000000dEaD;
156   address public marketingWallet = 0xD8Ad22Ae19340ff77373bE33970aE92883Aba80A;
157
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 157

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
156   address public marketingWallet = 0xD8Ad22Ae19340ff77373bE33970aE92883Aba80A;
157
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
161
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 158

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
157
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
161   struct Taxes {
162
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 159

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
161   struct Taxes {
162   uint256 rfi;
163
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 159

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
158   string private constant _name = "FROG CEO";
159   string private constant _symbol = "FROG CEO";
160
161   struct Taxes {
162   uint256 rfi;
163
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 162

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
161   struct Taxes {
162   uint256 rfi;
163   uint256 marketing;
164   }
165   // tax 6% r, 4% mrkt
166
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 162

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
161   struct Taxes {
162   uint256 rfi;
163   uint256 marketing;
164   }
165   // tax 6% r, 4% mrkt
166
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 272

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
271
272   function decreaseAllowance(address spender, uint256 subtractedValue)
273   public
274   returns (bool)
275   {
276
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 277

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
276   uint256 currentAllowance = _allowances[_msgSender()][spender];
277   require(currentAllowance >= subtractedValue, "BEP20: decreased allowance below
zero");
278   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
279
280   return true;
281
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 288

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
287
288   function isExcludedFromReward(address account) public view returns (bool) {
289   return _isExcluded[account];
290   }
291
292
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 318

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
317   if (_rOwned[account] > 0) {
318   _tOwned[account] = tokenFromReflection(_rOwned[account]);
319   }
320   _isExcluded[account] = true;
321   _excluded.push(account);
322
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 337

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
336
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 338

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341   function includeInFee(address account) public onlyOwner {
342
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 361

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
360   }
361   _rOwned[address(this)] += rMarketing;
362   }
363
364
365
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 361

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
360   }
361   _rOwned[address(this)] += rMarketing;
362   }
363
364
365
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 368

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
367   bool takeFee
368   ) private view returns (valuesFromGetValues memory to_return) {
369   to_return = _getTValues(tAmount, takeFee);
370   (
371   to_return.rAmount,
372
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 370

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
369   to_return = _getTValues(tAmount, takeFee);
370   (
371   to_return.rAmount,
372   to_return.rTransferAmount,
373   to_return.rRfi,
374
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 372

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
371   to_return.rAmount,
372   to_return.rTransferAmount,
373   to_return.rRfi,
374   to_return.rMarketing
375   ) = _getRValues(to_return, tAmount, takeFee, _getRate());
376
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 400

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
399   valuesFromGetValues memory s,
400   uint256 tAmount,
401   bool takeFee,
402   uint256 currentRate
403   )
404
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 400

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
399   valuesFromGetValues memory s,
400   uint256 tAmount,
401   bool takeFee,
402   uint256 currentRate
403   )
404
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 402

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
401   bool takeFee,
402   uint256 currentRate
403   )
404   private
405   pure
406
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 403

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
402   uint256 currentRate
403   )
404   private
405   pure
406   returns (
407
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
406   returns (
407   uint256 rAmount,
408   uint256 rTransferAmount,
409   uint256 rRfi,
410   uint256 rMarketing
411
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
406   returns (
407   uint256 rAmount,
408   uint256 rTransferAmount,
409   uint256 rRfi,
410   uint256 rMarketing
411
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 425

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
424   rMarketing;
425   return (rAmount, rTransferAmount, rRfi, rMarketing);
426   }
427
428   function _getRate() private view returns (uint256) {
429
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 429

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
428   function _getRate() private view returns (uint256) {
429   (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
430   return rSupply / tSupply;
431   }
432
433
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 429

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
428   function _getRate() private view returns (uint256) {
429   (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
430   return rSupply / tSupply;
431   }
432
433
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 433

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
432
433   function _getCurrentSupply() private view returns (uint256, uint256) {
434   uint256 rSupply = _rTotal;
435   uint256 tSupply = _tTotal;
436   for (uint256 i = 0; i < _excluded.length; i++) {
437
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 433

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
432
433   function _getCurrentSupply() private view returns (uint256, uint256) {
434   uint256 rSupply = _rTotal;
435   uint256 tSupply = _tTotal;
436   for (uint256 i = 0; i < _excluded.length; i++) {
437
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 437

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
436   for (uint256 i = 0; i < _excluded.length; i++) {
437   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
438   return (_rTotal, _tTotal);
439   rSupply = rSupply - _rOwned[_excluded[i]];
440   tSupply = tSupply - _tOwned[_excluded[i]];
441
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 442

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
441   }
442   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
443   return (rSupply, tSupply);
444   }
445
446
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 449

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
448   address spender,
449   uint256 amount
450   ) private {
451   require(owner != address(0), "BEP20: approve from the zero address");
452   require(spender != address(0), "BEP20: approve to the zero address");
453
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 451

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
450   ) private {
451   require(owner != address(0), "BEP20: approve from the zero address");
452   require(spender != address(0), "BEP20: approve to the zero address");
453   _allowances[owner][spender] = amount;
454   emit Approval(owner, spender, amount);
455
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 452

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
451   require(owner != address(0), "BEP20: approve from the zero address");
452   require(spender != address(0), "BEP20: approve to the zero address");
453   _allowances[owner][spender] = amount;
454   emit Approval(owner, spender, amount);
455   }
456
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 506

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
505   _rOwned[sender] = _rOwned[sender] - s.rAmount;
506   _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
507
508   if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
509   if (s.rMarketing > 0 || s.tMarketing > 0) _takeMarketing(s.rMarketing,
s.tMarketing);
510
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 509

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
508   if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
509   if (s.rMarketing > 0 || s.tMarketing > 0) _takeMarketing(s.rMarketing,
s.tMarketing);
510   emit Transfer(sender, recipient, s.tTransferAmount);
511   }
512
513
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 510

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
509   if (s.rMarketing > 0 || s.tMarketing > 0) _takeMarketing(s.rMarketing,
s.tMarketing);
510   emit Transfer(sender, recipient, s.tTransferAmount);
511   }
512
513   function swapAndLiquify() private lockTheSwap {
514
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 513

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
512
513   function swapAndLiquify() private lockTheSwap {
514
515   uint256 contractBalance = balanceOf(address(this));
516   swapTokensForBNB(contractBalance);
517
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 554

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
553
554   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
555   require(amount <= 1e15, "Cannot set swap threshold amount higher than 1% of
tokens");
556   swapTokensAtAmount = amount * 10**_decimals;
557   }
558
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
564
565   //Use this in case BEP20 Tokens are sent to the contract by mistake
566   function rescueAnyBEP20Tokens(address _tokenAddr,address _to, uint256 _amount)
public onlyOwner {
567   require(_tokenAddr != address(this), "Owner can't claim contract's balance of its
own tokens");
568   IBEP20(_tokenAddr).transfer(_to, _amount);
569
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
564
565   //Use this in case BEP20 Tokens are sent to the contract by mistake
566   function rescueAnyBEP20Tokens(address _tokenAddr,address _to, uint256 _amount)
public onlyOwner {
567   require(_tokenAddr != address(this), "Owner can't claim contract's balance of its
own tokens");
568   IBEP20(_tokenAddr).transfer(_to, _amount);
569
```

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

LINE 338

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341   function includeInFee(address account) public onlyOwner {
342
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 28

low SEVERITY

The current pragma Solidity directive is ""^0.8.17"". 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

- FROGCEO.sol

Locations

```
27
28   function allowance(address owner, address spender) external view returns (uint256);
29
30   function approve(address spender, uint256 amount) external returns (bool);
31
32
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 337

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
336
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 338

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341   function includeInFee(address account) public onlyOwner {
342
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 338

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
337   function excludeFromFee(address account) public onlyOwner {
338   _isExcludedFromFee[account] = true;
339   }
340
341   function includeInFee(address account) public onlyOwner {
342
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 443

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
442   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
443   return (rSupply, tSupply);
444   }
445
446   function _approve(
447
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 446

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
445
446   function _approve(
447   address owner,
448   address spender,
449   uint256 amount
450
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 450

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
449   uint256 amount
450   ) private {
451   require(owner != address(0), "BEP20: approve from the zero address");
452   require(spender != address(0), "BEP20: approve to the zero address");
453   _allowances[owner][spender] = amount;
454
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 451

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
450   ) private {
451   require(owner != address(0), "BEP20: approve from the zero address");
452   require(spender != address(0), "BEP20: approve to the zero address");
453   _allowances[owner][spender] = amount;
454   emit Approval(owner, spender, amount);
455
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 537

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
536   0, // accept any amount of ETH
537   path,
538   address(this),
539   block.timestamp
540   );
541
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 539

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
538   address(this),
539   block.timestamp
540   );
541   }
542
543
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 554

low SEVERITY

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

Source File

- FROGCEO.sol

Locations

```
553
554   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
555   require(amount <= 1e15, "Cannot set swap threshold amount higher than 1% of
tokens");
556   swapTokensAtAmount = amount * 10**_decimals;
557   }
558
```

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.