Bravo Arena

Smart Contract Audit Report

SYSFIXED

17 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Bravo Arena	BRV	Binance Smart Chain

Addresses

Contract address	0xEAc19378A08790ad1DAaD235fd33aDb8c314Ef07
Contract deployer address	0xe8260FbFE2e048D331c11b1b1dDCb812beEc1B34

Project Website

http://www.bravoarena.gg/

Codebase

https://bscscan.com/address/0xEAc19378A08790ad1DAaD235fd33aDb8c314Ef07#code

SUMMARY

BRAVO! A completely decentralized E-Sports platform on BSC where users can take part in various in-game tournaments. Play Warzone, CS:GO, Fortnite tournaments and win Crypto Prizes. Bravo has a house-edge of 10% of every contest, which goes to buybacks of the $BRV token.

Contract Summary

Documentation Quality

Bravo Arena 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 Bravo Arena 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 200, 222, 247, 276, 277, 406, 406, 407, 407, 408, 408, 409, 409, 439, 439, 469, 479, 490, 508, 519, 530, 548, 548, 555, 555, 562, 562, 569, 569, 576, 580, 580, 600, 601, 601, 603, 609, 610, 610, 611, 618, 618, 619, 619, 671, 671, 680, 680, 689, 689, 698, 698, 725, 738, 738, 739, 739, 740 and 740.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 13.
- 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 633, 634 and 726.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 508 and 710.

CONCLUSION

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

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

The issues found in the Bravo Arena 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 set, weak sources of randomness, and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value. We Recommend Don't use any of those environment variables as sources of randomness and being aware that the use of these variables introduces a certain level of trust in miners.

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.	ISSUE FOUND
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	Monday Jan 16 2023 17:19:56 GMT+0000 (Coordinated Universal Time)
Finished	Tuesday Jan 17 2023 18:38:49 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	BravoArena.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-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-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 200

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
199   require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
200   _approve(sender, _msgSender(), currentAllowance - amount);
201
202   return true;
203   }
204
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 222

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
221   {
222   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
223   return true;
224   }
225
226
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 247

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
246   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
247   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
248
249   return true;
250   }
251
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 276

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
275   require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
276   _balances[sender] = senderBalance - amount;
277   _balances[recipient] += amount;
278
279   emit Transfer(sender, recipient, amount);
280
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 277

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
276   _balances[sender] = senderBalance - amount;
277   _balances[recipient] += amount;
278
279   emit Transfer(sender, recipient, amount);
280   }
281
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 406

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
405
406   uint256 public tokenLiquidityThreshold = 75_000 * 10**decimals();
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 406

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
405
406   uint256 public tokenLiquidityThreshold = 75_000 * 10**decimals();
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
406   uint256 public tokenLiquidityThreshold = 75_000 * 10**decimals();
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
406   uint256 public tokenLiquidityThreshold = 75_000 * 10**decimals();
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 408

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411   uint256 public launchedAtBlock;
412
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 408

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
407   uint256 public maxBuyLimit = 1_000_000 * 10**decimals();
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411   uint256 public launchedAtBlock;
412
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 409

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411   uint256 public launchedAtBlock;
412
413
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 409

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
408   uint256 public maxSellLimit = 1_000_000 * 10**decimals();
409   uint256 public maxWalletLimit = 2_000_000 * 10**decimals();
410
411   uint256 public launchedAtBlock;
412
413
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
438   constructor() ERC20("Bravo Arena", "BRV") {
439   _tokengeneration(msg.sender, 100_000_000 * 10**decimals());
440   exemptFee[msg.sender] = true;
441
442   // IRouter _router = IRouter(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D); //
UNISWAP V2
443
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
438   constructor() ERC20("Bravo Arena", "BRV") {
439   _tokengeneration(msg.sender, 100_000_000 * 10**decimals());
440   exemptFee[msg.sender] = true;
441
442   // IRouter _router = IRouter(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D); //
UNISWAP V2
443
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 469

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
468   require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
469   _approve(sender, _msgSender(), currentAllowance - amount);
470
471   return true;
472   }
473
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
478   {
479   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
480   return true;
481   }
482
483
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 490

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
489   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
490   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
491
492   return true;
493   }
494
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 508

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
507
508   if(block.number < launchedAtBlock + 3 && sender == pair) {
509   nonCustodial[recipient] = true;
510   }
511
512
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 519

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
518   require(
519   balanceOf(recipient) + amount <= maxWalletLimit,
520   "You are exceeding maxWalletLimit"
521   );
522   }
523
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 530

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
529   require(
530   balanceOf(recipient) + amount <= maxWalletLimit,
531   "You are exceeding maxWalletLimit"
532   );
533   }
534
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 548

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
547   feeswap =
548   sellTaxes.liquidity +
549   sellTaxes.marketing +
550   sellTaxes.developer;
551   feesum = feeswap;
552
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 548

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
547   feeswap =
548   sellTaxes.liquidity +
549   sellTaxes.marketing +
550   sellTaxes.developer;
551   feesum = feeswap;
552
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 555

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
554   feeswap =
555   taxes.liquidity +
556   taxes.marketing +
557   taxes.developer ;
558   feesum = feeswap;
559
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 555

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
554   feeswap =
555   taxes.liquidity +
556   taxes.marketing +
557   taxes.developer ;
558   feesum = feeswap;
559
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 562

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
561   feeswap =
562   transferTaxes.liquidity +
563   transferTaxes.marketing +
564   transferTaxes.developer ;
565   feesum = feeswap;
566
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 562

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
561   feeswap =
562   transferTaxes.liquidity +
563   transferTaxes.marketing +
564   transferTaxes.developer ;
565   feesum = feeswap;
566
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 569

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
568
569   fee = (amount * feesum) / 100;
570
571   //send fees if threshold has been reached
572   //don't do this on buys, breaks swap
573
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 569

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
568
569   fee = (amount * feesum) / 100;
570
571   //send fees if threshold has been reached
572   //don't do this on buys, breaks swap
573
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 576

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
575   //rest to recipient
576   super._transfer(sender, recipient, amount - fee);
577   if (fee > 0) {
578   //send the fee to the contract
579   if (feeswap > 0) {
580
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 580

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
579   if (feeswap > 0) {
580   uint256 feeAmount = (amount * feeswap) / 100;
581   super._transfer(sender, address(this), feeAmount);
582   }
583
584
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 580

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
579   if (feeswap > 0) {
580   uint256 feeAmount = (amount * feeswap) / 100;
581   super._transfer(sender, address(this), feeAmount);
582   }
583
584
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 600

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
599   // Split the contract balance into halves
600   uint256 denominator = feeswap * 2;
601   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
602   denominator;
603   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
604
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 601

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
600   uint256 denominator = feeswap * 2;
601   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
602   denominator;
603   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
604
605
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 601

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
600   uint256 denominator = feeswap * 2;
601   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
602   denominator;
603   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
604
605
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 603

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
602   denominator;
603   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
604
605   uint256 initialBalance = address(this).balance;
606
607
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 609

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
608
609   uint256 deltaBalance = address(this).balance - initialBalance;
610   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
611   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
612
613
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 610

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
609   uint256 deltaBalance = address(this).balance - initialBalance;
610   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
611   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
612
613   if (ethToAddLiquidityWith > 0) {
614
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 610

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
609   uint256 deltaBalance = address(this).balance - initialBalance;
610   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
611   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
612
613   if (ethToAddLiquidityWith > 0) {
614
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 611

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
610   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
611   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
612
613   if (ethToAddLiquidityWith > 0) {
614   // Add liquidity to pancake
615
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 618

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
617
618   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
619   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
620   if (marketingAmt > 0) {
621   payable(marketingWallet).sendValue(marketingAmt);
622
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 618

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
617
618   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
619   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
620   if (marketingAmt > 0) {
621   payable(marketingWallet).sendValue(marketingAmt);
622
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 619

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
618   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
619   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
620   if (marketingAmt > 0) {
621   payable(marketingWallet).sendValue(marketingAmt);
622   }
623
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 619

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
618   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
619   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
620   if (marketingAmt > 0) {
621   payable(marketingWallet).sendValue(marketingAmt);
622   }
623
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 671

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
670   require(new_amount <= 1_000_000 && new_amount > 0, "Swap threshold amount should be
lower or euqal to 1% of tokens");
671   tokenLiquidityThreshold = new_amount * 10**decimals();
672   }
673
674   function SetBuyTaxes(
675
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 671

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
670   require(new_amount <= 1_000_000 && new_amount > 0, "Swap threshold amount should be
lower or euqal to 1% of tokens");
671   tokenLiquidityThreshold = new_amount * 10**decimals();
672   }
673
674   function SetBuyTaxes(
675
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 680

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
679   taxes = Taxes(_marketing, _liquidity,  _developer);
680   require((_marketing + _liquidity +  _developer) <= 10, "Must keep fees at 10% or
less");
681   }
682
683   function SetSellTaxes(
684
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 680

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
679   taxes = Taxes(_marketing, _liquidity,  _developer);
680   require((_marketing + _liquidity +  _developer) <= 10, "Must keep fees at 10% or
less");
681   }
682
683   function SetSellTaxes(
684
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 689

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
688   sellTaxes = Taxes(_marketing, _liquidity,  _developer);
689   require((_marketing + _liquidity + _developer) <= 10, "Must keep fees at 10% or
less");
690   }
691
692   function SetTransferTaxes(
693
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 689

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
688   sellTaxes = Taxes(_marketing, _liquidity,  _developer);
689   require((_marketing + _liquidity + _developer) <= 10, "Must keep fees at 10% or
less");
690   }
691
692   function SetTransferTaxes(
693
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 698

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
697   transferTaxes = Taxes(_marketing, _liquidity,  _developer);
698   require((_marketing + _liquidity + _developer) <= 10, "Must keep fees at 10% or
less");
699   }
700
701   function updateRouterAndPair(address newRouter, address newPair) external onlyOwner
{
702
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 698

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
697   transferTaxes = Taxes(_marketing, _liquidity,  _developer);
698   require((_marketing + _liquidity + _developer) <= 10, "Must keep fees at 10% or
less");
699   }
700
701   function updateRouterAndPair(address newRouter, address newPair) external onlyOwner
{
702
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 725

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
724   function bulkExemptFee(address[] memory accounts, bool state) external onlyOwner {
725   for (uint256 i = 0; i < accounts.length; i++) {
726   exemptFee[accounts[i]] = state;
727   }
728   }
729
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 738

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
737   require(maxWallet >= 500_000, "Cannot set max wallet amount lower than 0.5%");
738   maxBuyLimit = maxBuy * 10**decimals();
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 738

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
737   require(maxWallet >= 500_000, "Cannot set max wallet amount lower than 0.5%");
738   maxBuyLimit = maxBuy * 10**decimals();
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 739

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
738   maxBuyLimit = maxBuy * 10**decimals();
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
743
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 739

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
738   maxBuyLimit = maxBuy * 10**decimals();
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
743
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 740

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
743   function rescueETH(uint256 weiAmount) external onlyOwner {
744
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 740

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
739   maxSellLimit = maxSell * 10**decimals();
740   maxWalletLimit = maxWallet * 10**decimals();
741   }
742
743   function rescueETH(uint256 weiAmount) external onlyOwner {
744
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 13

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

- BravoArena.sol

Locations

```
12
13   pragma solidity ^0.8.17;
14
15   abstract contract Context {
16   function _msgSender() internal view virtual returns (address) {
17
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 633

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
632   address[] memory path = new address[](2);
633   path[0] = address(this);
634   path[1] = router.WETH();
635
636   _approve(address(this), address(router), tokenAmount);
637
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 634

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
633   path[0] = address(this);
634   path[1] = router.WETH();
635
636   _approve(address(this), address(router), tokenAmount);
637
638
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 726

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
725   for (uint256 i = 0; i < accounts.length; i++) {
726   exemptFee[accounts[i]] = state;
727   }
728   }
729
730
```

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

LINE 508

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
507
508   if(block.number < launchedAtBlock + 3 && sender == pair) {
509   nonCustodial[recipient] = true;
510   }
511
512
```

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

LINE 710

low SEVERITY

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

Source File

- BravoArena.sol

Locations

```
709   providingLiquidity = true;
710   launchedAtBlock = block.number;
711   }
712
713   function updateWallets(address _marketingWallet, address _devWallet) external
onlyOwner {
714
```

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.