Borzoi Inu

Smart Contract Audit Report

SYSFIXED

24 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Borzoi Inu	BORZ	Binance Smart Chain

Addresses

Contract address	0x08da9eb6147694e671a455d946a620a70d721eae
Contract deployer address	0xe5946E00E18Ad1C00c901e082b2E6A2D077699A1

Project Website

https://borztoken.com/

Codebase

https://bscscan.com/address/0x08da9eb6147694e671a455d946a620a70d721eae#contracts

SUMMARY

Decentralized wolfhound living on the Binance Smart Chain. Long Snout Watch automatic price tracking platform for all projects on BSC, Ethereum & Polygon.

Contract Summary

Documentation Quality

Borzoi Inu 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 Borzoi Inu 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 27, 31, 35, 39, 45, 52, 344, 344, 354, 354, 355, 355, 468, 498, 532, 532, 533, 534, 535, 538, 538, 543, 546, 546, 547, 547, 571, 572, 577, 583, 583, 584, 586, 587, 588, 593, 599, 599, 600, 602, 603, 604 and 609.
- 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 555 and 556.

CONCLUSION

We have audited the Borzoi Inu project released on January 2023 to discover issues and identify potential security vulnerabilities in Borzoi Inu 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 Borzoi Inu 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 and out-of-bounds array access. The index access expression can cause an exception in case of an invalid array index value.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	PASS
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
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	Monday Jan 23 2023 16:25:02 GMT+0000 (Coordinated Universal Time)
Finished	Tuesday Jan 24 2023 06:04:45 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	BORZ.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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 27

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
26   function add(uint256 a, uint256 b) internal pure returns (uint256) {
27   return a + b;
28   }
29
30   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
31
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 31

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
30   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
31   return a - b;
32   }
33
34   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
35
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 35

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
34   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
35   return a * b;
36   }
37
38   function div(uint256 a, uint256 b) internal pure returns (uint256) {
39
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 39

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
38   function div(uint256 a, uint256 b) internal pure returns (uint256) {
39   return a / b;
40   }
41
42   function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
43
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 45

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
44   require(b <= a, errorMessage);
45   return a - b;
46   }
47   }
48
49
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 52

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
51   require(b > 0, errorMessage);
52   return a / b;
53   }
54   }
55
56
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 344

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
343   uint8 private constant _supplyDecimals = 9;
344   uint256 private _tokenSupply = 1000000000 * 10**_supplyDecimals;
345
346   // Buy/Sell Tax
347   uint256 public Tax_Buy = 5;
348
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 344

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
343   uint8 private constant _supplyDecimals = 9;
344   uint256 private _tokenSupply = 1000000000 * 10**_supplyDecimals;
345
346   // Buy/Sell Tax
347   uint256 public Tax_Buy = 5;
348
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
353
354   uint256 public _BagLimit = _tokenSupply * 7 / 100;
355   uint256 public _TransactionLimit = _tokenSupply * 7 / 100;
356
357   // Swap Trigger & Transaction Counter
358
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
353
354   uint256 public _BagLimit = _tokenSupply * 7 / 100;
355   uint256 public _TransactionLimit = _tokenSupply * 7 / 100;
356
357   // Swap Trigger & Transaction Counter
358
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 355

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
354   uint256 public _BagLimit = _tokenSupply * 7 / 100;
355   uint256 public _TransactionLimit = _tokenSupply * 7 / 100;
356
357   // Swap Trigger & Transaction Counter
358   uint8 private tx_Counter = 0;
359
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 355

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
354   uint256 public _BagLimit = _tokenSupply * 7 / 100;
355   uint256 public _TransactionLimit = _tokenSupply * 7 / 100;
356
357   // Swap Trigger & Transaction Counter
358   uint8 private tx_Counter = 0;
359
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 468

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
467   uint256 bagSize = balanceOf(to);
468   require((bagSize + amount) <= _BagLimit, "Error: bag limit reached.");
469   }
470
471   if (from != owner())
472
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 498

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
497   }
498   tx_Counter++;
499   }
500   _tokenTransfer(from, to, amount, feeUsed, isBuy);
501   }
502
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 532

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
531   // Burn Split
532   uint256 tokens_to_Burn = contractTokenBalance * SplitBurn / 100;
533   _tokenSupply = _tokenSupply - tokens_to_Burn;
534   _balances[walletDEAD] = _balances[walletDEAD] + tokens_to_Burn;
535   _balances[address(this)] = _balances[address(this)] - tokens_to_Burn;
536
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 532

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
531   // Burn Split
532   uint256 tokens_to_Burn = contractTokenBalance * SplitBurn / 100;
533   _tokenSupply = _tokenSupply - tokens_to_Burn;
534   _balances[walletDEAD] = _balances[walletDEAD] + tokens_to_Burn;
535   _balances[address(this)] = _balances[address(this)] - tokens_to_Burn;
536
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 533

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
532   uint256 tokens_to_Burn = contractTokenBalance * SplitBurn / 100;
533   _tokenSupply = _tokenSupply - tokens_to_Burn;
534   _balances[walletDEAD] = _balances[walletDEAD] + tokens_to_Burn;
535   _balances[address(this)] = _balances[address(this)] - tokens_to_Burn;
536
537
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 534

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
533   _tokenSupply = _tokenSupply - tokens_to_Burn;
534   _balances[walletDEAD] = _balances[walletDEAD] + tokens_to_Burn;
535   _balances[address(this)] = _balances[address(this)] - tokens_to_Burn;
536
537   // Fee Split
538
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 535

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
534   _balances[walletDEAD] = _balances[walletDEAD] + tokens_to_Burn;
535   _balances[address(this)] = _balances[address(this)] - tokens_to_Burn;
536
537   // Fee Split
538   uint256 tokensMarketing = contractTokenBalance * SplitMarketing / 100;
539
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 538

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
537   // Fee Split
538   uint256 tokensMarketing = contractTokenBalance * SplitMarketing / 100;
539
540   // Swap for BNB
541   uint256 balanceBeforeSwap = address(this).balance;
542
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 538

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
537   // Fee Split
538   uint256 tokensMarketing = contractTokenBalance * SplitMarketing / 100;
539
540   // Swap for BNB
541   uint256 balanceBeforeSwap = address(this).balance;
542
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 543

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
542   swapTokensForBNB(tokensMarketing);
543   uint256 TotalBNB = address(this).balance - balanceBeforeSwap;
544
545   // Marketing Split
546   uint256 MarketingSize = SplitMarketing * 100 / SplitMarketing;
547
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 546

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
545   // Marketing Split
546   uint256 MarketingSize = SplitMarketing * 100 / SplitMarketing;
547   uint256 MarketingBNB = TotalBNB * MarketingSize / 100;
548
549   sendToWallet(walletMarketing, MarketingBNB);
550
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 546

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
545   // Marketing Split
546   uint256 MarketingSize = SplitMarketing * 100 / SplitMarketing;
547   uint256 MarketingBNB = TotalBNB * MarketingSize / 100;
548
549   sendToWallet(walletMarketing, MarketingBNB);
550
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 547

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
546   uint256 MarketingSize = SplitMarketing * 100 / SplitMarketing;
547   uint256 MarketingBNB = TotalBNB * MarketingSize / 100;
548
549   sendToWallet(walletMarketing, MarketingBNB);
550   }
551
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 547

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
546   uint256 MarketingSize = SplitMarketing * 100 / SplitMarketing;
547   uint256 MarketingBNB = TotalBNB * MarketingSize / 100;
548
549   sendToWallet(walletMarketing, MarketingBNB);
550   }
551
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 571

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
570
571   _balances[sender] = _balances[sender] - tokenAmount;
572   _balances[recipient] = _balances[recipient] + tokenAmount;
573
574   emit Transfer(sender, recipient, tokenAmount);
575
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
571   _balances[sender] = _balances[sender] - tokenAmount;
572   _balances[recipient] = _balances[recipient] + tokenAmount;
573
574   emit Transfer(sender, recipient, tokenAmount);
575
576
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 577

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
576   if (recipient == walletDEAD)
577   _tokenSupply = _tokenSupply - tokenAmount;
578
579
580
581
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 583

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
582
583   uint256 BuyFee = tokenAmount * Tax_Buy/100;
584   uint256 taxedTokenAmount = tokenAmount - BuyFee;
585
586   _balances[sender] = _balances[sender] - tokenAmount;
587
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 583

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
582
583   uint256 BuyFee = tokenAmount * Tax_Buy/100;
584   uint256 taxedTokenAmount = tokenAmount - BuyFee;
585
586   _balances[sender] = _balances[sender] - tokenAmount;
587
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 584

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
583   uint256 BuyFee = tokenAmount * Tax_Buy/100;
584   uint256 taxedTokenAmount = tokenAmount - BuyFee;
585
586   _balances[sender] = _balances[sender] - tokenAmount;
587   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
588
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 586

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
585
586   _balances[sender] = _balances[sender] - tokenAmount;
587   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
588   _balances[address(this)] = _balances[address(this)] + BuyFee;
589
590
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 587

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
586   _balances[sender] = _balances[sender] - tokenAmount;
587   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
588   _balances[address(this)] = _balances[address(this)] + BuyFee;
589
590   emit Transfer(sender, recipient, taxedTokenAmount);
591
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 588

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
587   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
588   _balances[address(this)] = _balances[address(this)] + BuyFee;
589
590   emit Transfer(sender, recipient, taxedTokenAmount);
591
592
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 593

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
592   if (recipient == walletDEAD)
593   _tokenSupply = _tokenSupply - taxedTokenAmount;
594
595
596
597
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 599

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
598
599   uint256 SellFee = tokenAmount * Tax_Sell/100;
600   uint256 taxedTokenAmount = tokenAmount - SellFee;
601
602   _balances[sender] = _balances[sender] - tokenAmount;
603
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 599

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
598
599   uint256 SellFee = tokenAmount * Tax_Sell/100;
600   uint256 taxedTokenAmount = tokenAmount - SellFee;
601
602   _balances[sender] = _balances[sender] - tokenAmount;
603
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 600

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
599   uint256 SellFee = tokenAmount * Tax_Sell/100;
600   uint256 taxedTokenAmount = tokenAmount - SellFee;
601
602   _balances[sender] = _balances[sender] - tokenAmount;
603   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
604
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 602

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
601
602   _balances[sender] = _balances[sender] - tokenAmount;
603   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
604   _balances[address(this)] = _balances[address(this)] + SellFee;
605
606
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 603

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
602   _balances[sender] = _balances[sender] - tokenAmount;
603   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
604   _balances[address(this)] = _balances[address(this)] + SellFee;
605
606   emit Transfer(sender, recipient, taxedTokenAmount);
607
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 604

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
603   _balances[recipient] = _balances[recipient] + taxedTokenAmount;
604   _balances[address(this)] = _balances[address(this)] + SellFee;
605
606   emit Transfer(sender, recipient, taxedTokenAmount);
607
608
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 609

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
608   if (recipient == walletDEAD)
609   _tokenSupply = _tokenSupply - taxedTokenAmount;
610
611   }
612   }
613
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 555

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
554   address[] memory path = new address[](2);
555   path[0] = address(this);
556   path[1] = uniswapV2Router.WETH();
557   _approve(address(this), address(uniswapV2Router), tokenAmount);
558   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
559
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 556

low SEVERITY

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

Source File

- BORZ.sol

Locations

```
555   path[0] = address(this);
556   path[1] = uniswapV2Router.WETH();
557   _approve(address(this), address(uniswapV2Router), tokenAmount);
558   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
559   tokenAmount,
560
```

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.