



Zeuswap

Smart Contract Audit Report

TABLE OF CONTENTS

Audited Details

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

Summary

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

Conclusion

Audit Results

Smart Contract Analysis

- Detected Vulnerabilities

Disclaimer

About Us

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Zeuswap	ZEUS	Binance Smart Chain

Addresses

Contract address	0x321427ec9759e535d1033d5C0478D4bEA9a0CB56
Contract deployer address	0x169155527F850C3A3c897e153F7e61cE43b72f66

Project Website

<https://www.zeuswap.finance/>

Codebase

<https://bscscan.com/address/0x321427ec9759e535d1033d5C0478D4bEA9a0CB56#code>

SUMMARY

ZEUSWAP is the fastest & most secure DEX with double-layer SSL protection and the first ever with Insurance Protection Scheme to ensure users' assets protection. With our Insurance Program, our goal is to put the days of getting rugged in DeFi a thing of the past. Our Advantages are AUDIT + KYC, DAPP + SWAP LIVE: <https://dapp.zeuswap.finance/swap>, No private sales, Buyback tax to ensure stable token price, AMA & Calls with Tier 1 Channels, High APR Staking, 200% extra token for BIG BUY BONUS.

Contract Summary

Documentation Quality

Zeuswap 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 Zeuswap 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 568, 578, 586, 605, 607, 619, 620, 634, 636, 749, 749, 749, 750, 750, 750, 766, 766, 767, 803, 803, 803, 805, 815, 815, 815, 817, 865, 869, 870, 874, 874, 899, 899, 900, 909, 915, 916, 931, 959, 961, 962, 963, 964, 964, 967, 967, 972, 972, 977 and 977.
- 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 921, 922, 949 and 950.

CONCLUSION

We have audited the NamaFile project released on January 2023 to discover issues and identify potential security vulnerabilities in NamaFile 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 NamaFile 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 issue found is out of bounds array access be aware 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.	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

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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 568

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
567     unchecked {  
568         _approve(sender, _msgSender(), currentAllowance - amount);  
569     }  
570 }  
571  
572
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
577 function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
578     _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
579     return true;
580 }
581
582
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 586

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
585     unchecked {
586         _approve(_msgSender(), spender, currentAllowance - subtractedValue);
587     }
588
589     return true;
590
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 605

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
604 unchecked {  
605   _balances[sender] = senderBalance - amount;  
606 }  
607 _balances[recipient] += amount;  
608  
609
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 607

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
606     }  
607     _balances[recipient] += amount;  
608  
609     emit Transfer(sender, recipient, amount);  
610  
611
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 619

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
618
619  _totalSupply += amount;
620  _balances[account] += amount;
621  emit Transfer(address(0), account, amount);
622
623
```


SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 620

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
619  _totalSupply += amount;  
620  _balances[account] += amount;  
621  emit Transfer(address(0), account, amount);  
622  
623  _afterTokenTransfer(address(0), account, amount);  
624
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 634

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
633     unchecked {
634         _balances[account] = accountBalance - amount;
635     }
636     _totalSupply -= amount;
637
638
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 636

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
635     }
636     _totalSupply -= amount;
637
638     emit Transfer(account, address(0), amount);
639
640
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 749

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
748
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 749

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
748
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 749

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
748
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 750

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell   = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
754
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 750

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell   = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
754
```


SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 750

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
749  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
750  _totalFeesOnSell   = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
751
752  walletToWalletTransferFee = 0;
753
754
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 766

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
765
766   _mint(owner(), 1e8 * (10 ** decimals()));
767   swapTokensAtAmount = totalSupply() / 5000;
768   }
769
770
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 766

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
765
766   _mint(owner(), 1e8 * (10 ** decimals()));
767   swapTokensAtAmount = totalSupply() / 5000;
768   }
769
770
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 767

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
766  _mint(owner(), 1e8 * (10 ** decimals()));
767  swapTokensAtAmount = totalSupply() / 5000;
768  }
769
770  receive() external payable {
771
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 803

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
802
803  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
804
805  require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
806
807
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 803

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
802
803  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
804
805  require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
806
807
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 803

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
802
803  _totalFeesOnBuy    = liquidityFeeOnBuy + marketingFeeOnBuy + insuranceFeeOnBuy +
devFeeOnBuy;
804
805  require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
806
807
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 805

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
804
805   require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
806         25%");
807   emit UpdateBuyFees(liquidityFeeOnBuy, marketingFeeOnBuy, insuranceFeeOnBuy,
808         devFeeOnBuy);
809 }
```


SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 815

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
814 devFeeOnSell      = _devFeeOnSell;
815 _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
816
817 require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
818
819
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 815

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
814 devFeeOnSell      = _devFeeOnSell;
815 _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
816
817 require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
818
819
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 815

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
814 devFeeOnSell      = _devFeeOnSell;
815 _totalFeesOnSell  = liquidityFeeOnSell + marketingFeeOnSell + insuranceFeeOnSell +
devFeeOnSell;
816
817 require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
25%");
818
819
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 817

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
816
817   require(_totalFeesOnBuy + _totalFeesOnSell <= 25, "Total Fees cannot be more than
818         25%");
819   emit UpdateSellFees(liquidityFeeOnSell, marketingFeeOnSell, insuranceFeeOnSell,
820         devFeeOnSell);
821 }
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 865

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
864     to == uniswapV2Pair &&
865     _totalFeesOnBuy + _totalFeesOnSell > 0
866     ) {
867         swapping = true;
868
869
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 869

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
868
869  uint256 totalFee = _totalFeesOnBuy + _totalFeesOnSell;
870  uint256 liquidityShare = liquidityFeeOnBuy + liquidityFeeOnSell;
871
872
873
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 870

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
869  uint256 totalFee = _totalFeesOnBuy + _totalFeesOnSell;
870  uint256 liquidityShare = liquidityFeeOnBuy + liquidityFeeOnSell;
871
872
873  if (liquidityShare > 0) {
874
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 874

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
873     if (liquidityShare > 0) {  
874         uint256 liquidityTokens = (contractTokenBalance * liquidityShare) / totalFee;  
875         swapAndLiquify(liquidityTokens);  
876     }  
877  
878
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 874

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
873     if (liquidityShare > 0) {  
874         uint256 liquidityTokens = (contractTokenBalance * liquidityShare) / totalFee;  
875         swapAndLiquify(liquidityTokens);  
876     }  
877  
878
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 899

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
898     if (_totalFees > 0) {
899         uint256 fees = (amount * _totalFees) / 100;
900         amount = amount - fees;
901         super._transfer(from, address(this), fees);
902     }
903 
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 899

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
898     if (_totalFees > 0) {
899         uint256 fees = (amount * _totalFees) / 100;
900         amount = amount - fees;
901         super._transfer(from, address(this), fees);
902     }
903
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 900

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
899     uint256 fees = (amount * _totalFees) / 100;
900     amount = amount - fees;
901     super._transfer(from, address(this), fees);
902 }
903
904
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 909

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
908     function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
909         require(newAmount > totalSupply() / 1000000, "SwapTokensAtAmount must be greater
than 0.0001% of total supply");
910         swapTokensAtAmount = newAmount;
911         emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
912     }
913
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 915

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
914 function swapAndLiquify(uint256 tokens) private {
915     uint256 half = tokens / 2;
916     uint256 otherHalf = tokens - half;
917
918     uint256 initialBalance = address(this).balance;
919 }
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 916

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
915  uint256 half = tokens / 2;  
916  uint256 otherHalf = tokens - half;  
917  
918  uint256 initialBalance = address(this).balance;  
919  
920
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 931

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
930
931     uint256 newBalance = address(this).balance - initialBalance;
932
933     uniswapV2Router.addLiquidityETH{value: newBalance}(
934         address(this),
935
```


SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 959

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
958
959  uint256 newBalance = address(this).balance - initialBalance;
960
961  uint256 marketingShare = marketingFeeOnBuy + marketingFeeOnSell;
962  uint256 insuranceShare = insuranceFeeOnBuy + insuranceFeeOnSell;
963
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 961

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
960
961  uint256 marketingShare = marketingFeeOnBuy + marketingFeeOnSell;
962  uint256 insuranceShare = insuranceFeeOnBuy + insuranceFeeOnSell;
963  uint256 devShare      = devFeeOnBuy + devFeeOnSell;
964  uint256 totalShare = marketingShare + insuranceShare + devShare;
965
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 962

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
961 uint256 marketingShare = marketingFeeOnBuy + marketingFeeOnSell;  
962 uint256 insuranceShare = insuranceFeeOnBuy + insuranceFeeOnSell;  
963 uint256 devShare      = devFeeOnBuy + devFeeOnSell;  
964 uint256 totalShare = marketingShare + insuranceShare + devShare;  
965  
966
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 963

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
962 uint256 insuranceShare = insuranceFeeOnBuy + insuranceFeeOnSell;  
963 uint256 devShare       = devFeeOnBuy + devFeeOnSell;  
964 uint256 totalShare = marketingShare + insuranceShare + devShare;  
965  
966 if (marketingShare > 0) {  
967
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 964

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
963 uint256 devShare      = devFeeOnBuy + devFeeOnSell;
964 uint256 totalShare = marketingShare + insuranceShare + devShare;
965
966 if (marketingShare > 0) {
967     uint256 marketingTokens = (newBalance * marketingShare) / totalShare;
968 }
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 964

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
963 uint256 devShare          = devFeeOnBuy + devFeeOnSell;
964 uint256 totalShare = marketingShare + insuranceShare + devShare;
965
966 if (marketingShare > 0) {
967     uint256 marketingTokens = (newBalance * marketingShare) / totalShare;
968 }
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 967

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
966   if (marketingShare > 0) {
967       uint256 marketingTokens = (newBalance * marketingShare) / totalShare;
968       payable(marketingWallet).sendValue(marketingTokens);
969   }
970
971
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 967

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
966   if (marketingShare > 0) {
967       uint256 marketingTokens = (newBalance * marketingShare) / totalShare;
968       payable(marketingWallet).sendValue(marketingTokens);
969   }
970
971
```


SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 972

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
971  if (insuranceShare > 0) {  
972  uint256 insuranceTokens = (newBalance * insuranceShare) / totalShare;  
973  payable(insuranceWallet).sendValue(insuranceTokens);  
974  }  
975  
976
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 972

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
971     if (insuranceShare > 0) {
972         uint256 insuranceTokens = (newBalance * insuranceShare) / totalShare;
973         payable(insuranceWallet).sendValue(insuranceTokens);
974     }
975
976
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 977

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
976   if (devShare > 0) {  
977       uint256 devTokens = (newBalance * devShare) / totalShare;  
978       payable(devWallet).sendValue(devTokens);  
979   }  
980  
981
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 977

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
976     if (devShare > 0) {
977         uint256 devTokens = (newBalance * devShare) / totalShare;
978         payable(devWallet).sendValue(devTokens);
979     }
980
981
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 921

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
920     address[] memory path = new address[](2);
921     path[0] = address(this);
922     path[1] = uniswapV2Router.WETH();
923
924     uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
925
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 922

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
921 path[0] = address(this);
922 path[1] = uniswapV2Router.WETH();
923
924 uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
925     half,
926
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 949

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
948 address[] memory path = new address[](2);
949 path[0] = address(this);
950 path[1] = uniswapV2Router.WETH();
951
952 uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
953
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 950

low SEVERITY

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

Source File

- Zeuswap.sol

Locations

```
949 path[0] = address(this);
950 path[1] = uniswapV2Router.WETH();
951
952 uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
953     tokenAmount,
954
```


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.