

eVault Smart Contract Audit Report



11 Dec 2022



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	
eVault	EVLT	Binance Smart Chain	

Addresses

Contract address	0xeE34Af939a75223571ac818f0958b67cba48cd01
Contract deployer address	0x7F8C481f359Bb7fcb09aCf6a4a67a28F8FE37995

Project Website

https://www.evaultproject.com/

<u>Codebase</u>

https://bscscan.com/address/0xeE34Af939a75223571ac818f0958b67cba48cd01#code



SUMMARY

One of the primary issues with DeFi right now is the need for users to utilize multiple different platforms to cover the full range of DeFi Services. The eVault Platform will solve that problem by serving as an integrated All-In-One (AIO) DeFi platform. The eVault Platform integrates many different DeFi services, including a staking platform, a launchpad, DEX, P2P swaps, and other utilities as we evolve. The eVault Platform is in advanced stages of development and will go live in Q2 of 2023.

Contract Summary

Documentation Quality

eVault 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 eVault with the discovery of several low issues.

Test Coverage

Test coverage of the project is 100% (Through Codebase)

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 825, 834 and 840.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 579, 602, 635, 638, 660, 663, 689, 691, 741, 861, 880, 912, 913, 914, 915, 930, 931, 932, 933, 994, 997, 1000, 1005, 1008, 1015, 1016, 1017, 1018, 1019, 1022, 1023 and 1024.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 7, 27, 125, 174, 201, 286, 371, 401 and 791.
- SWC-110 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 814, 817, 853, 855, 868, 873, 917, 935, 1014, 1021, 1036 and 1037.



CONCLUSION

We have audited the eVault project released on December 2022 to discover issues and identify potential security vulnerabilities in eVault 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 code on eVault smart contract do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, a state variable visibility is not set, a public state variable with array type causing reachable exception by default and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.



AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	ISSUE FOUND
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
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
Assert Violation	SWC-110	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	Delegate calls 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
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
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





SMART CONTRACT ANALYSIS

Started	Saturday Dec 10 2022 03:44:20 GMT+0000 (Coordinated Universal Time)		
Finished	Sunday Dec 11 2022 17:12:41 GMT+0000 (Coordinated Universal Time)		
Mode	Standard		
Main Source File	EVAULTTOKEN.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-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged



SYSFIXED

SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.	low	acknowledged
SWC-110	PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged



SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 579

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
578 address owner = _msgSender();
579 _approve(owner, spender, allowance(owner, spender) + addedValue);
580 return true;
581 }
582 583
```



SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 602

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
601 unchecked {
602 _approve(owner, spender, currentAllowance - subtractedValue);
603 }
604
605 return true;
606
```



SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 635

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
634 unchecked {
635 _balances[from] = fromBalance - amount;
636 // Overflow not possible: the sum of all balances is capped by totalSupply, and the
sum is preserved by
637 // decrementing then incrementing.
638 _balances[to] += amount;
639
```



SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 638

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
637 // decrementing then incrementing.
638 _balances[to] += amount;
639 }
640
641 emit Transfer(from, to, amount);
642
```



SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 660

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
659
660 _totalSupply += amount;
661 unchecked {
662 // Overflow not possible: balance + amount is at most totalSupply + amount, which
is checked above.
663 _balances[account] += amount;
664
```



SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 663

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
662 // Overflow not possible: balance + amount is at most totalSupply + amount, which
is checked above.
663 _balances[account] += amount;
664 }
665 emit Transfer(address(0), account, amount);
666
667
```



SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 689

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
688 unchecked {
689 _balances[account] = accountBalance - amount;
690 // Overflow not possible: amount <= accountBalance <= totalSupply.
691 _totalSupply -= amount;
692 }
693</pre>
```



SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 691

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
690 // Overflow not possible: amount <= accountBalance <= totalSupply.
691 _totalSupply -= amount;
692 }
693
694 emit Transfer(account, address(0), amount);
695
```



SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 741

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
740 unchecked {
741 __approve(owner, spender, currentAllowance - amount);
742 }
743 }
744 }
745
```



SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 861

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
860 stakingWallet = payable(_staking);
861 __mint(msg.sender, _supply * 10 ** decimals());
862 excluded[marketingWallet] = true;
863 excluded[msg.sender] = true;
864 excluded[devWallet] = true;
865
```



SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 861

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
860 stakingWallet = payable(_staking);
861 __mint(msg.sender, _supply * 10 ** decimals());
862 excluded[marketingWallet] = true;
863 excluded[msg.sender] = true;
864 excluded[devWallet] = true;
865
```



SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 880

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
879 sellFeeAmt = 80;
880 swapTreshold = _supply * 10 ** decimals() / 1_000_000; // 0.0001% of the supply
881
882 }
883
884
```



SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 880

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
879 sellFeeAmt = 80;
880 swapTreshold = _supply * 10 ** decimals() / 1_000_000; // 0.0001% of the supply
881
882 }
883
884
```



SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 880

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
879 sellFeeAmt = 80;
880 swapTreshold = _supply * 10 ** decimals() / 1_000_000; // 0.0001% of the supply
881
882 }
883
884
```



SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 912

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
911 function setBuyFee(uint32 _marketing, uint32 _dev, uint32 _staking) external
onlyOwner {
912 uint32 total = _marketing + _dev + _staking;
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
```



SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 912

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
911 function setBuyFee(uint32 _marketing, uint32 _dev, uint32 _staking) external
onlyOwner {
912 uint32 total = _marketing + _dev + _staking;
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
```



SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 913

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
912 uint32 total = _marketing + _dev + _staking;
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917
```



SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 913

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
912 uint32 total = _marketing + _dev + _staking;
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917
```



SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 914

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917 fees[0] = FeeStruct({
918
```



SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 914

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
913 uint32 marketingPercent = (_marketing * 1000) / total;
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917 fees[0] = FeeStruct({
918
```



SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 915

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917 fees[0] = FeeStruct({
918 marketing: marketingPercent,
919
```



SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 915

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
914 uint32 devPercent = (_dev * 1000) / total;
915 uint32 stakingPercent = (_staking * 1000) / total;
916
917 fees[0] = FeeStruct({
918 marketing: marketingPercent,
919
```



SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 930

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
929 function setSellFee(uint32 _marketing, uint32 _dev, uint32 _staking) external
onlyOwner {
930 uint32 total = _marketing + _dev + _staking;
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
```



SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 930

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
929 function setSellFee(uint32 _marketing, uint32 _dev, uint32 _staking) external
onlyOwner {
930 uint32 total = _marketing + _dev + _staking;
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
```



SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 931

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
930 uint32 total = _marketing + _dev + _staking;
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935
```


LINE 931

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
930 uint32 total = _marketing + _dev + _staking;
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935
```



LINE 932

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935 fees[1] = FeeStruct({
936
```



LINE 932

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
931 uint32 marketingPercent = (_marketing * 1000) / total;
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935 fees[1] = FeeStruct({
936
```



LINE 933

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935 fees[1] = FeeStruct({
936 marketing: marketingPercent,
937
```



LINE 933

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
932 uint32 devPercent = (_dev * 1000) / total;
933 uint32 stakingPercent = (_staking * 1000) / total;
934
935 fees[1] = FeeStruct({
936 marketing: marketingPercent,
937
```



LINE 994

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
993 if (_isSell) {
994 taxAmt = (_amount * sellFeeAmt) / 1000;
995 }
996 else {
997 taxAmt = (_amount * buyFeeAmt) / 1000;
998
```



LINE 994

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
993 if (_isSell) {
994 taxAmt = (_amount * sellFeeAmt) / 1000;
995 }
996 else {
997 taxAmt = (_amount * buyFeeAmt) / 1000;
998
```



LINE 997

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
996 else {
997 taxAmt = (_amount * buyFeeAmt) / 1000;
998 buyFeesCollected = taxAmt;
999 }
1000 uint remaining = _amount - taxAmt;
1001
```



LINE 997

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
996 else {
997 taxAmt = (_amount * buyFeeAmt) / 1000;
998 buyFeesCollected = taxAmt;
999 }
1000 uint remaining = _amount - taxAmt;
1001
```



LINE 1000

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
999 }
1000 uint remaining = _amount - taxAmt;
1001 super._transfer(_from, address(this), taxAmt);
1002 super._transfer(_from, _to, remaining);
1003 }
1004
```



LINE 1005

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

1004 function processTax() internal swapLocked() {
1005 uint buyShare = (buyFeesCollected * 1e8) / balanceOf(address(this));
1006 uint balanceBefore = address(this).balance;
1007 swapForBnb(balanceOf(address(this)));
1008 uint swapped = address(this).balance - balanceBefore;
1009



LINE 1005

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

1004 function processTax() internal swapLocked() {
1005 uint buyShare = (buyFeesCollected * 1e8) / balanceOf(address(this));
1006 uint balanceBefore = address(this).balance;
1007 swapForBnb(balanceOf(address(this)));
1008 uint swapped = address(this).balance - balanceBefore;
1009



LINE 1008

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

1007 swapForBnb(balanceOf(address(this))); 1008 uint swapped = address(this).balance - balanceBefore; 1009 uint toDev; 1010 uint toMarketing; 1011 uint toStaking; 1012



LINE 1015

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1014 current = fees[0];
1015 uint buyCut = (swapped * buyShare) / 1e8;
1016 toDev = (buyCut * current.dev) / 1000;
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019
```



LINE 1015

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1014 current = fees[0];
1015 uint buyCut = (swapped * buyShare) / 1e8;
1016 toDev = (buyCut * current.dev) / 1000;
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019
```



LINE 1016

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

1015 uint buyCut = (swapped * buyShare) / 1e8; 1016 toDev = (buyCut * current.dev) / 1000; 1017 toMarketing = (buyCut * current.marketing) / 1000; 1018 toStaking = (buyCut - toMarketing - toDev); 1019 swapped -= buyCut; 1020



LINE 1016

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

1015 uint buyCut = (swapped * buyShare) / 1e8; 1016 toDev = (buyCut * current.dev) / 1000; 1017 toMarketing = (buyCut * current.marketing) / 1000; 1018 toStaking = (buyCut - toMarketing - toDev); 1019 swapped -= buyCut; 1020



LINE 1017

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1016 toDev = (buyCut * current.dev) / 1000;
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019 swapped -= buyCut;
1020 }
1021
```



LINE 1017

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1016 toDev = (buyCut * current.dev) / 1000;
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019 swapped -= buyCut;
1020 }
1021
```



LINE 1018

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019 swapped -= buyCut;
1020 }
1021 current = fees[1];
1022
```



LINE 1018

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018 toStaking = (buyCut - toMarketing - toDev);
1019 swapped -= buyCut;
1020 }
1021 current = fees[1];
1022
```



LINE 1019

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1018 toStaking = (buyCut - toMarketing - toDev);
1019 swapped -= buyCut;
1020 }
1021 current = fees[1];
1022 toDev += (swapped * current.dev) / 1000;
1023
```



LINE 1022

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1021 current = fees[1];
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026
```



LINE 1022

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1021 current = fees[1];
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026
```



LINE 1022

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1021 current = fees[1];
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026
```



LINE 1023

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027
```



LINE 1023

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027
```



LINE 1023

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027
```



LINE 1024

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027 (bool staking,) = stakingWallet.call{value: toStaking}("");
1028
```



LINE 1024

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027 (bool staking,) = stakingWallet.call{value: toStaking}("");
1028
```



LINE 1024

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025 (bool dev,) = devWallet.call{value: toDev}("");
1026 (bool market,) = marketingWallet.call{value: toMarketing}("");
1027 (bool staking,) = stakingWallet.call{value: toStaking}("");
1028
```



LINE 7

Iow SEVERITY

The current pragma Solidity directive is "">=0.5.0"". 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

- EVAULTTOKEN.sol

```
6
7 pragma solidity >=0.5.0;
8
9 interface IUniswapV2Factory {
10 event PairCreated(address indexed token0, address indexed token1, address pair, uint);
11
```



LINE 27

Iow SEVERITY

The current pragma Solidity directive is "">=0.6.2"". 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

- EVAULTTOKEN.sol

```
26
27 pragma solidity >=0.6.2;
28
29 interface IUniswapV2Router01 {
30 function factory() external pure returns (address);
31
```





LINE 125

Iow SEVERITY

The current pragma Solidity directive is "">=0.6.2"". 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

- EVAULTTOKEN.sol

```
124
125 pragma solidity >=0.6.2;
126
127
128 interface IUniswapV2Router02 is IUniswapV2Router01 {
129
```



LINE 174

IOW SEVERITY

The current pragma Solidity directive is ""^0.8.0"". 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

- EVAULTTOKEN.sol

Locations

173
174 pragma solidity ^0.8.0;
175
176 /**
177 * @dev Provides information about the current execution context, including the
178



LINE 201

Iow SEVERITY

The current pragma Solidity directive is ""^0.8.0"". 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

- EVAULTTOKEN.sol

Locations

200 201 pragma solidity ^0.8.0; 202 203 204 /** 205


LINE 286

Iow SEVERITY

The current pragma Solidity directive is ""^0.8.0"". 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

- EVAULTTOKEN.sol

Locations

285
286 pragma solidity ^0.8.0;
287
288 /**
289 * @dev Interface of the ERC20 standard as defined in the EIP.
290



LINE 371

Iow SEVERITY

The current pragma Solidity directive is ""^0.8.0"". 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

- EVAULTTOKEN.sol

Locations

370
371 pragma solidity ^0.8.0;
372
373
374 /**
375



LINE 401

Iow SEVERITY

The current pragma Solidity directive is ""^0.8.0"". 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

- EVAULTTOKEN.sol

Locations

400 401 pragma solidity ^0.8.0; 402 403 404 405



LINE 791

Iow SEVERITY

The current pragma Solidity directive is ""^0.8"". 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

- EVAULTTOKEN.sol

Locations

790 791 pragma solidity ^0.8; 792 793 794 795





C

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 825

Iow SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "buyFeesCollected" is internal. Other possible visibility settings are public and private.

Source File

- EVAULTTOKEN.sol

Locations

824
825 uint256 buyFeesCollected;
826
827 // The wierd ordering is for var packing. Each time we skip a line we change slots.
828 /// @notice The address that collects the marketing tax
829



SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 834

Iow SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwap" is internal. Other possible visibility settings are public and private.

Source File

- EVAULTTOKEN.sol

```
833 address payable public devWallet;
834 bool inSwap;
835
836 /// @notice The address that collects the staking tax
837 address payable public stakingWallet;
838
```



SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 840

Iow SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "wbnb" is internal. Other possible visibility settings are public and private.

Source File

- EVAULTTOKEN.sol

Locations

839
840 address wbnb;
841
842 modifier swapLocked() {
843 inSwap = true;
844





SWC-110 | PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.

LINE 814

Iow SEVERITY

The public state variable "fees" in "EVAULTTOKEN" contract has type "struct EVAULTTOKEN.FeeStruct[2]" and can cause an exception in case of use of invalid array index value.

Source File

- EVAULTTOKEN.sol

Locations

813 /// @notice The fee structure repartition. Index 0 contains the buy fee, index 1
contains the sell fees.
814 FeeStruct[2] public fees;
815
816 /// @notice All the routers of the supported DEXes
817 IUniswapV2Router02[] public routers;



SWC-110 | PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.

LINE 817

Iow SEVERITY

The public state variable "routers" in "EVAULTTOKEN" contract has type "contract IUniswapV2Router02[]" and can cause an exception in case of use of invalid array index value.

Source File

- EVAULTTOKEN.sol

Locations

816 /// @notice All the routers of the supported DEXes
817 IUniswapV2Router02[] public routers;
818
819 /// @notice The router used by the contract for swapping the taxes
820 IUniswapV2Router02 public preferredRouter;



LINE 853

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
852 _approve(address(this), 0x10ED43C718714eb63d5aA57B78B54704E256024E,
type(uint256).max);
853 address firstPair =
IUniswapV2Factory(routers[0].factory()).createPair(address(this), routers[0].WETH());
854 isLP[firstPair] = true;
855 wbnb = routers[0].WETH();
856
857
```





LINE 853

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
852 _approve(address(this), 0x10ED43C718714eb63d5aA57B78B54704E256024E,
type(uint256).max);
853 address firstPair =
IUniswapV2Factory(routers[0].factory()).createPair(address(this), routers[0].WETH());
854 isLP[firstPair] = true;
855 wbnb = routers[0].WETH();
856
857
```





LINE 855

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
854 isLP[firstPair] = true;
855 wbnb = routers[0].WETH();
856
857 // Sets addresses and mints tokens to owner.
858 marketingWallet = payable(_marketing);
859
```



LINE 868

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

867 // Sets fees 868 fees[0] = FeeStruct({ 869 marketing: 400, 870 dev: 200, 871 staking: 400 872



LINE 873

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
872 });
873 fees[1] = FeeStruct({
874 marketing: 375,
875 dev: 250,
876 staking: 375
877
```



LINE 917

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

916 917 fees[0] = FeeStruct({ 918 marketing: marketingPercent, 919 dev: devPercent, 920 staking: stakingPercent 921



LINE 935

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

Locations

934 935 fees[1] = FeeStruct({ 936 marketing: marketingPercent, 937 dev: devPercent, 938 staking: stakingPercent 939



LINE 1014

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1013 if(buyShare > 0) {
1014 current = fees[0];
1015 uint buyCut = (swapped * buyShare) / 1e8;
1016 toDev = (buyCut * current.dev) / 1000;
1017 toMarketing = (buyCut * current.marketing) / 1000;
1018
```



LINE 1021

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1020 }
1021 current = fees[1];
1022 toDev += (swapped * current.dev) / 1000;
1023 toMarketing += (swapped * current.marketing) / 1000;
1024 toStaking += (swapped * current.staking) / 1000;
1025
```



LINE 1036

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1035 address[] memory path = new address[](2);
1036 path[0] = address(this);
1037 path[1] = wbnb;
1038
1039 // make the swap
1040
```



LINE 1037

Iow SEVERITY

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

Source File

- EVAULTTOKEN.sol

```
1036 path[0] = address(this);
1037 path[1] = wbnb;
1038
1039 // make the swap
1040 preferredRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
1041
```



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.