swipe2earn.com Token

Smart Contract Audit Report

SYSFIXED

23 Sep 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
swipe2earn.com Token	SWIPE	BSC

Addresses

Contract address	0x57C8Ed6E4333C051E36EdFD115fed274f737d497
Contract deployer address	0xDBB46F70578cc2053d02F7BA140afBbf919E7636

Project Website

https://www.swipe2earn.com/

Codebase

https://bscscan.com/address/0x57C8Ed6E4333C051E36EdFD115fed274f737d497#contracts

SUMMARY

The easiest way to earn crypto: swipe2earn rewards you for watching sponsored content. Our advantage is the doxxed team, german company, audit, passive income, earning crypto by watching videos, hyper burn mechanism, and listing on pancakeswap right after the sale. The CEO Julian günther has a lot of experience in managing startups. He founded fitgun, which is one of the largest sellers of sporting goods in europe. The whole swipe2earn team is based in Germany.

Contract Summary

Documentation Quality

swipe2earn.com Token provides a document with a very good standard of solidity base code.

- The technical description is provided clearly and structured and also don't have any risk issue.

Code Quality

The Overall quality of the basecode is GOOD

- Standart solidity basecode and rules are already followed with swipe2earn.com Token Project .

Test Coverage

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

Audit Findings Summary

- SWC-101 | Arithmetic operation Issues discovered on lines 325, 348, 381, 383, 404, 405, 430, 481, 634, 648, 663, 664, 677, 689, 704, 718, 732, 746, 762, 785, 808, 834, 1271, 1272, 1274, 1296, 1357, 1364, 1405, 1407, 1471, 1495, 1500, 1505, and 1407.
- SWC-103 | A floating pragma is set on lines 10, 94, 122, 148, 531, 614, 843, and 1065. The current pragma Solidity directive is ""^0.8.17"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.
- SWC-108 | State variable visibility is not set on lines 1293 .It is best practice to set the visibility of state variables explicitly. The default visibility for "protections" is internal. Other possible visibility settings are public and private.
- SWC-110 | Out of bounds array access on lines 1406, 1407, 1472, 1473, 1474, 1613 and 1614.

CONCLUSION

We have audited the swipe2earn.com Token Coin which has released on September 2022 to discover issues and identify potential security vulnerabilities in swipe2earn.com Token Project. This process is used to find bugs, technical issues, and security loopholes that find some common issues in the code.

The security audit report produced satisfactory results with a low risk issue on the contract project.

The most common issue found in writing code on contracts that do not pose a big risk, writing on contracts is close to the standard of writing contracts in general. Some of the low issues that were found stated variable visibility are not set, a floating pragma is set and out of bounds array access The index access expression can cause an exception in case of 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
Check-Effect Interaction	SWC-107	Check-Effect-Interaction pattern should be followed if the code performs ANY external 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 Caller	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
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	Thu Sep 10 2022 08:14:04 GMT+0000 (Coordinated Universal Time)
Finished	Fri Sep 11 2022 09:10:04 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	SWIPE.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	COMPILER-REWRITABLE "<UINT> - 1" 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
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-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 325

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
324   address owner = _msgSender();
325   _approve(owner, spender, allowance(owner, spender) + addedValue);
326   return true;
327   }
328
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 348

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
347   unchecked {
348   _approve(owner, spender, currentAllowance - subtractedValue);
349   }
350
351   return true;
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 381

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
380   unchecked {
381   _balances[from] = fromBalance - amount;
382   }
383   _balances[to] += amount;
384
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 383

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
382   }
383   _balances[to] += amount;
384
385   emit Transfer(from, to, amount);
386
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 404

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
403
404   _totalSupply += amount;
405   _balances[account] += amount;
406   emit Transfer(address(0), account, amount);
407
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 405

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
404   _totalSupply += amount;
405   _balances[account] += amount;
406   emit Transfer(address(0), account, amount);
407
408   _afterTokenTransfer(address(0), account, amount);
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 430

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
429   unchecked {
430   _balances[account] = accountBalance - amount;
431   }
432   _totalSupply -= amount;
433
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 432

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
431   }
432   _totalSupply -= amount;
433
434   emit Transfer(account, address(0), amount);
435
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 481

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
480   unchecked {
481   _approve(owner, spender, currentAllowance - amount);
482   }
483   }
484   }
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 634

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
633   unchecked {
634   uint256 c = a + b;
635   if (c < a) return (false, 0);
636   return (true, c);
637   }
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 648

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
647   if (b > a) return (false, 0);
648   return (true, a - b);
649   }
650   }
651
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 663

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
662   if (a == 0) return (true, 0);
663   uint256 c = a * b;
664   if (c / a != b) return (false, 0);
665   return (true, c);
666   }
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 664

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
663   uint256 c = a * b;
664   if (c / a != b) return (false, 0);
665   return (true, c);
666   }
667   }
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 677

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
676   if (b == 0) return (false, 0);
677   return (true, a / b);
678   }
679   }
680
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 689

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
688   if (b == 0) return (false, 0);
689   return (true, a % b);
690   }
691   }
692
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 704

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
703   function add(uint256 a, uint256 b) internal pure returns (uint256) {
704   return a + b;
705   }
706
707   /**
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 718

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
717   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
718   return a - b;
719   }
720
721   /**
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 732

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
731   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
732   return a * b;
733   }
734
735   /**
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 746

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
745   function div(uint256 a, uint256 b) internal pure returns (uint256) {
746   return a / b;
747   }
748
749   /**
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 762

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
761   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
762   return a % b;
763   }
764
765   /**
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 785

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
784   require(b <= a, errorMessage);
785   return a - b;
786   }
787   }
788
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 808

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
807   require(b > 0, errorMessage);
808   return a / b;
809   }
810   }
811
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 834

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
833   require(b > 0, errorMessage);
834   return a % b;
835   }
836   }
837   }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1271

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1270   uint256 private constant MAX = ~uint256(0);
1271   uint256 private _tTotal = 5000000000 * 10**18;
1272   uint256 private _rTotal = (MAX - (MAX % _tTotal));
1273   uint256 private _tFeeTotal;
1274   uint256 private _burnLimit = 100000000 * 10**18;
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1272

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1271   uint256 private _tTotal = 5000000000 * 10**18;
1272   uint256 private _rTotal = (MAX - (MAX % _tTotal));
1273   uint256 private _tFeeTotal;
1274   uint256 private _burnLimit = 100000000 * 10**18;
1275
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1274

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1273   uint256 private _tFeeTotal;
1274   uint256 private _burnLimit = 100000000 * 10**18;
1275
1276   uint8 private _decimals = 18;
1277
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1296

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1295
1296   uint256 private numTokensSellToAddToLiquidity = 500000 * 10**18;
1297
1298   event MinTokensBeforeSwapUpdated(uint256 minTokensBeforeSwap);
1299   event SwapAndLiquifyEnabledUpdated(bool enabled);
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1357

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1356
1357   uint256 _amount = tAmount.mul(10**18);
1358   address account = _msgSender();
1359   uint256 current_balance = balanceOf( account );
1360
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1364

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1363
1364   if( (_tTotal > _burnLimit) && (_tTotal - _amount) < _burnLimit) _amount =
_tTotal.sub(_burnLimit);
1365   require(_tTotal > _burnLimit,"cannot burn more, final supply 100,000,000");
1366
1367   uint256 rAmount = _amount.mul(_getRate());
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1405

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1404   require(_isExcluded[account], "Account is already included");
1405   for (uint256 i = 0; i < _excluded.length; i++) {
1406   if (_excluded[i] == account) {
1407   _excluded[i] = _excluded[_excluded.length - 1];
1408   _tOwned[account] = 0;
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1407

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1406   if (_excluded[i] == account) {
1407   _excluded[i] = _excluded[_excluded.length - 1];
1408   _tOwned[account] = 0;
1409   _isExcluded[account] = false;
1410   _excluded.pop();
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1471

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1470   uint256 tSupply = _tTotal;
1471   for (uint256 i = 0; i < _excluded.length; i++) {
1472   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1473   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1474   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1495

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1494   return _amount.mul(_taxFee).div(
1495   10**3
1496   );
1497   }
1498   function calculateDevelopmentFee(uint256 _amount) private view returns (uint256) {
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1500

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1499   return _amount.mul(_developmentFee).div(
1500   10**3
1501   );
1502   }
1503   function calculateMarketingFee(uint256 _amount) private view returns (uint256) {
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1505

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1504   return _amount.mul(_marketingFee).div(
1505   10**3
1506   );
1507   }
1508   function removeAllFee() private {
```

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

LINE 1407

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1406   if (_excluded[i] == account) {
1407   _excluded[i] = _excluded[_excluded.length - 1];
1408   _tOwned[account] = 0;
1409   _isExcluded[account] = false;
1410   _excluded.pop();
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 10

low 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

- SWIPE.Sol

Locations

```
9
10   pragma solidity ^0.8.0;
11
12   /**
13   * @dev Interface of the ERC20 standard as defined in the EIP.
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 94

low 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

- SWIPE.Sol

Locations

```
93
94   pragma solidity ^0.8.0;
95
96   /**
97   * @dev Interface for the optional metadata functions from the ERC20 standard.
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 122

low 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

- SWIPE.Sol

Locations

```
121
122   pragma solidity ^0.8.0;
123
124   /**
125   * @dev Provides information about the current execution context, including the
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 148

low 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

- SWIPE.Sol

Locations

```
147
148   pragma solidity ^0.8.0;
149
150
151
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 531

low 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

- SWIPE.Sol

Locations

```
530
531   pragma solidity ^0.8.0;
532
533   /**
534   * @dev Contract module which provides a basic access control mechanism, where
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 614

low 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

- SWIPE.Sol

Locations

```
613
614   pragma solidity ^0.8.0;
615
616   // CAUTION
617   // This version of SafeMath should only be used with Solidity 0.8 or later,
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 843

low SEVERITY

The current pragma Solidity directive is ""^0.8.1"". 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

- SWIPE.Sol

Locations

```
842
843   pragma solidity ^0.8.1;
844
845   /**
846   * @dev Collection of functions related to the address type
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 1065

low SEVERITY

The current pragma Solidity directive is ""^0.8.4"". 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

- SWIPE.Sol

Locations

```
1064
1065   pragma solidity ^0.8.4;
1066
1067
1068
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 1293

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1292   address public immutable uniswapV2Pair;
1293   bool inSwapAndLiquify;
1294   bool public swapAndLiquifyEnabled = true;
1295
1296   uint256 private numTokensSellToAddToLiquidity = 500000 * 10**18;
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1406

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1405   for (uint256 i = 0; i < _excluded.length; i++) {
1406   if (_excluded[i] == account) {
1407   _excluded[i] = _excluded[_excluded.length - 1];
1408   _tOwned[account] = 0;
1409   _isExcluded[account] = false;
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1407

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1406   if (_excluded[i] == account) {
1407   _excluded[i] = _excluded[_excluded.length - 1];
1408   _tOwned[account] = 0;
1409   _isExcluded[account] = false;
1410   _excluded.pop();
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1472

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1471   for (uint256 i = 0; i < _excluded.length; i++) {
1472   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1473   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1474   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1475   }
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1473

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1472   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1473   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1474   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1475   }
1476   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1474

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1473   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1474   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1475   }
1476   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
1477   return (rSupply, tSupply);
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1613

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1612   address[] memory path = new address[](2);
1613   path[0] = address(this);
1614   path[1] = uniswapV2Router.WETH();
1615   _approve(address(this), address(uniswapV2Router), tokenAmount);
1616   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1614

low SEVERITY

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

Source File

- SWIPE.Sol

Locations

```
1613   path[0] = address(this);
1614   path[1] = uniswapV2Router.WETH();
1615   _approve(address(this), address(uniswapV2Router), tokenAmount);
1616   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1617   tokenAmount,
```

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.