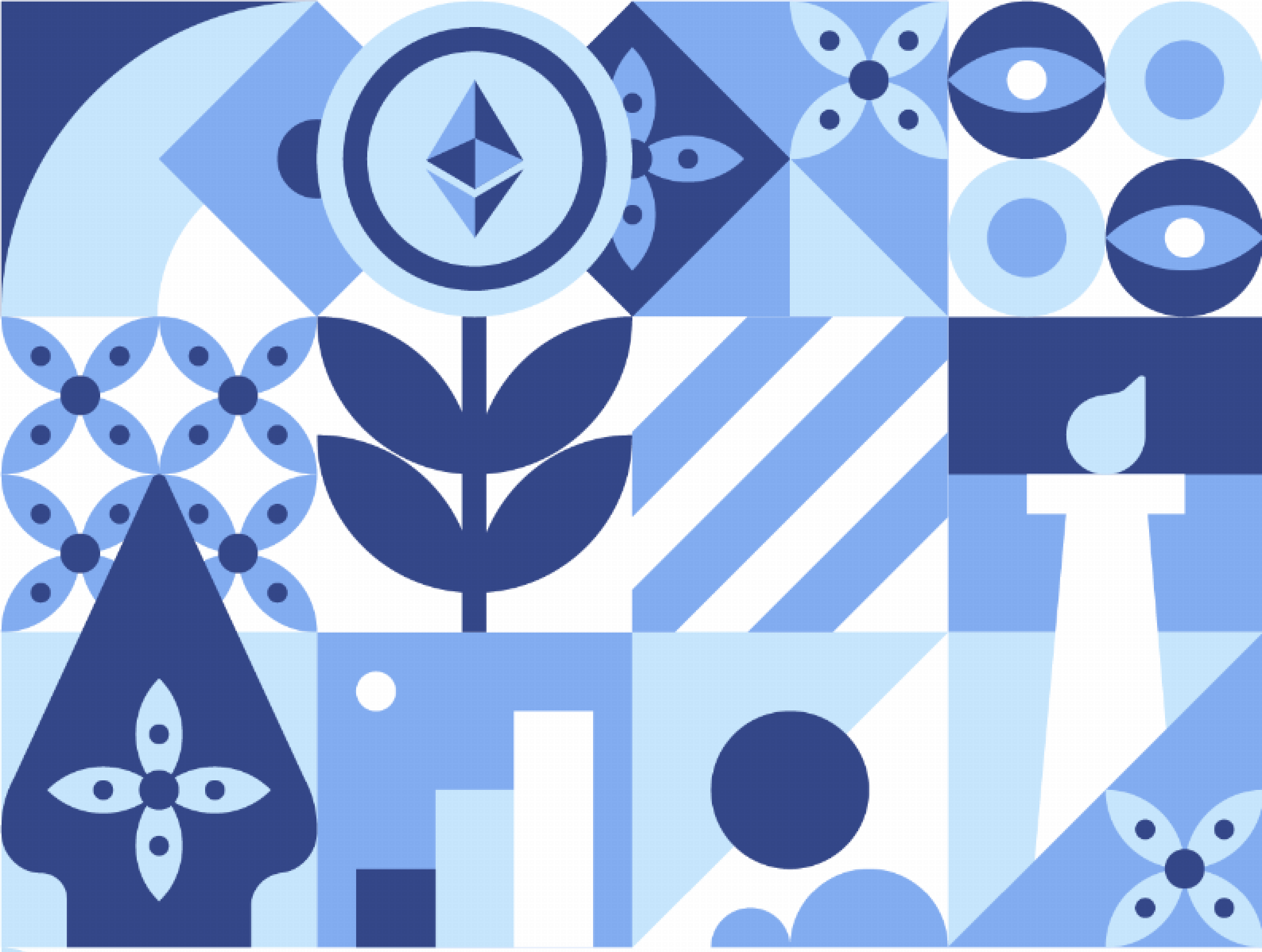

Streamer Inu

Smart Contract Audit Report

SYSFIXED

24 Feb 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Streamer Inu	STREAMERINU	Ethereum

Addresses

Contract address	0x0e8d2eb7d6bdf28393c25a1966385ad32ff0259a
Contract deployer address	0xDb80EEc593836e8042a018Bc1B6D8614aF584091

Project Website

https://streamerinu.com/

Codebase

https://etherscan.io/address/0x0e8d2eb7d6bdf28393c25a1966385ad32ff0259a#code

SUMMARY

We here at $Streamerinu have one purpose – to help streamers' dreams come true. As fellow streamers who never really made it, we were so inspired by the effort of others in the Defi space to help spread joy and positivity across the streaming community, and we felt it was our duty to jump in and keep pushing that energy.

Contract Summary

Documentation Quality

Streamer Inu provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also dont have any high risk issue.

Code Quality

The Overall quality of the basecode is standard.

- Standard solidity basecode and rules are already followed by Streamer Inu with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 401, 417, 442, 472, 475, 493, 497, 521, 525, 620, 636, 649, 649, 663, 676, 690, 704, 717, 732, 747, 770, 793, 823, 1150, 1151, 1153, 1156, 1156, 1160, 1160, 1164, 1164, 1214, 1214, 1219, 1219, 1229, 1229, 1229, 1231, 1232, 1237, 1237, 1237, 1241, 1241, 1262, 1262, 1273, 1273, 1374, 1391, 1420, 1435, 1435, 1435, 1438, 1438, 1438, 1438, 1439, 1439, 1442, 1442, 1442, 1445, 1446, 1446, 1447, 1449, 1449, 1456, 1496, 1496, 1504, 1505, 1509, 1509, 1509, 1526, 1526 and 1592.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 20.
- 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 1464 and 1464.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1354 and 1361.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1355 and 1362.

CONCLUSION

We have audited the Streamer Inu project released on February 2022 to discover issues and identify potential security vulnerabilities in Streamer Inu Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

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

The issues found in the Streamer Inu smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, weak sources of randomness, tx.origin as a part of authorization control, and out-of-bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value. We recommend avoiding Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
Unprotected Ether Withdrawal	SWC-105	Due to missing or insufficient access controls, malicious parties can withdraw from the contract.	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recursive call.	PASS
Uninitialized Storage Pointer	SWC-109	Uninitialized local storage variables can point to unexpected storage locations in the contract.	PASS
Assert Violation	SWC-110 SWC-123	Properly functioning code should never reach a failing assert statement.	ISSUE FOUND
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Callee	SWC-112	Delegatecalls should only be allowed to trusted addresses.	PASS

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	ISSUE FOUND
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	ISSUE FOUND
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations.	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS
Insufficient Gas Griefing	SWC-126	Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract.	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.	PASS

Typographical Error	SWC-129	A typographical error can occur for example when the intent of a defined operation is to sum a number to a variable.	PASS
Override control character	SWC-130	Malicious actors can use the Right-To-Left-Override unicode character to force RTL text rendering and confuse users as to the real intent of a contract.	PASS
Unused variables	SWC-131 SWC-135	Unused variables are allowed in Solidity and they do not pose a direct security issue.	PASS
Unexpected Ether balance	SWC-132	Contracts can behave erroneously when they strictly assume a specific Ether balance.	PASS
Hash Collisions Variable	SWC-133	Using abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision.	PASS
Hardcoded gas amount	SWC-134	The transfer() and send() functions forward a fixed amount of 2300 gas.	PASS
Unencrypted Private Data	SWC-136	It is a common misconception that private type variables cannot be read.	PASS

SMART CONTRACT ANALYSIS

Started	Wednesday Feb 23 2022 13:15:00 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Feb 24 2022 15:54:58 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	StreamerInu.sol

Detected Issues

ID	Title	Severity	Status
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-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-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	low	acknowledged
SWC-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 401

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
400    *
401    * This is an alternative to {approve} that can be used as a mitigation for
402    * problems described in {IERC20-approve}.
403    *
404    * Emits an {Approval} event indicating the updated allowance.
405
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 417

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
416    * @dev Atomically decreases the allowance granted to `spender` by the caller.
417    *
418    * This is an alternative to {approve} that can be used as a mitigation for
419    * problems described in {IERC20-approve}.
420    *
421
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 442

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
441   *
442   * This internal function is equivalent to {transfer}, and can be used to
443   * e.g. implement automatic token fees, slashing mechanisms, etc.
444   *
445   * Emits a {Transfer} event.
446
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 472

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
471
472   _afterTokenTransfer(sender, recipient, amount);
473   }
474
475   /** @dev Creates `amount` tokens and assigns them to `account`, increasing
476
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
474
475   /** @dev Creates `amount` tokens and assigns them to `account`, increasing
476   * the total supply.
477   *
478   * Emits a {Transfer} event with `from` set to the zero address.
479
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 493

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
492
493   _afterTokenTransfer(address(0), account, amount);
494   }
495
496   /**
497
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 497

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
496   /**
497   * @dev Destroys `amount` tokens from `account`, reducing the
498   * total supply.
499   *
500   * Emits a {Transfer} event with `to` set to the zero address.
501
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 521

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
520
521    _afterTokenTransfer(account, address(0), amount);
522    }
523
524    /**
525
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 525

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
524   /**
525   * @dev Sets `amount` as the allowance of `spender` over the `owner` s tokens.
526   *
527   * This internal function is equivalent to `approve`, and can be used to
528   * e.g. set automatic allowances for certain subsystems, etc.
529
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 620

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
619   /**
620   * @dev Returns the substraction of two unsigned integers, with an overflow flag.
621   *
622   * _Available since v3.4._
623   */
624
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 636

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
635   */
636   function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
637   unchecked {
638   // Gas optimization: this is cheaper than requiring 'a' not being zero, but the
639   // benefit is lost if 'b' is also tested.
640
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 649

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
648    /**
649    * @dev Returns the division of two unsigned integers, with a division by zero flag.
650    *
651    * _Available since v3.4._
652    */
653
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 649

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
648   /**
649   * @dev Returns the division of two unsigned integers, with a division by zero flag.
650   *
651   * _Available since v3.4._
652   */
653
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 663

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
662   *
663   * _Available since v3.4._
664   */
665   function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
666   unchecked {
667
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 676

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
675    *
676    * Counterpart to Solidity's `+` operator.
677    *
678    * Requirements:
679    *
680
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 690

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
689   *
690   * Counterpart to Solidity's `-` operator.
691   *
692   * Requirements:
693   *
694
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 704

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
703    *
704    * Counterpart to Solidity's `*` operator.
705    *
706    * Requirements:
707    *
708
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 717

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
716    * division by zero. The result is rounded towards zero.
717    *
718    * Counterpart to Solidity's `/` operator.
719    *
720    * Requirements:
721
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 732

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
731    *
732    * Counterpart to Solidity's `%` operator. This function uses a `revert`
733    * opcode (which leaves remaining gas untouched) while Solidity uses an
734    * invalid opcode to revert (consuming all remaining gas).
735    *
736
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 747

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
746   * overflow (when the result is negative).
747   *
748   * CAUTION: This function is deprecated because it requires allocating memory for
the error
749   * message unnecessarily. For custom revert reasons use {trySub}.
750   *
751
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 770

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
769    * @dev Returns the integer division of two unsigned integers, reverting with custom message on
770    * division by zero. The result is rounded towards zero.
771    *
772    * Counterpart to Solidity's `/` operator. Note: this function uses a
773    * `revert` opcode (which leaves remaining gas untouched) while Solidity
774
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 793

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
792    * @dev Returns the remainder of dividing two unsigned integers. (unsigned integer modulo),
793    * reverting with custom message when dividing by zero.
794    *
795    * CAUTION: This function is deprecated because it requires allocating memory for the error
796    * message unnecessarily. For custom revert reasons use {tryMod}.
797
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 823

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
822   interface IUniswapV2Factory {
823   event PairCreated(
824   address indexed token0,
825   address indexed token1,
826   address pair,
827
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1150

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1149   maxTransactionAmount = 10_000_000 * 1e18; // 1% from total supply
maxTransactionAmountTxn
1150   maxWallet = 20_000_000 * 1e18; // 2% from total supply maxWallet
1151   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1152
1153   buyMarketingFee = _buyMarketingFee;
1154
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1151

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1150   maxWallet = 20_000_000 * 1e18; // 2% from total supply maxWallet
1151   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1152
1153   buyMarketingFee = _buyMarketingFee;
1154   buyLiquidityFee = _buyLiquidityFee;
1155
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1153

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1152
1153   buyMarketingFee = _buyMarketingFee;
1154   buyLiquidityFee = _buyLiquidityFee;
1155   buyDevFee = _buyDevFee;
1156   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1157
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1156

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1155   buyDevFee = _buyDevFee;
1156   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1157
1158   sellMarketingFee = _sellMarketingFee;
1159   sellLiquidityFee = _sellLiquidityFee;
1160
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1156

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1155   buyDevFee = _buyDevFee;
1156   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1157
1158   sellMarketingFee = _sellMarketingFee;
1159   sellLiquidityFee = _sellLiquidityFee;
1160
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1160

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1159   sellLiquidityFee = _sellLiquidityFee;
1160   sellDevFee = _sellDevFee;
1161   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1162
1163   marketingWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as
marketing wallet
1164
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1160

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1159   sellLiquidityFee = _sellLiquidityFee;
1160   sellDevFee = _sellDevFee;
1161   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1162
1163   marketingWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as marketing wallet
1164
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1164

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1163   marketingWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as marketing wallet
1164   devWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as dev wallet
1165
1166   // exclude from paying fees or having max transaction amount
1167   excludeFromFees(owner(), true);
1168
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1164

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1163   marketingWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as marketing wallet
1164   devWallet = address(0xBf80197478BeA72047DE1C8B88Add8426f1D833a); // set as dev wallet
1165
1166   // exclude from paying fees or having max transaction amount
1167   excludeFromFees(owner(), true);
1168
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1214

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1213   require(
1214   newAmount <= (totalSupply() * 5) / 1000,
1215   "Swap amount cannot be higher than 0.5% total supply."
1216   );
1217   swapTokensAtAmount = newAmount;
1218
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1214

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1213   require(
1214   newAmount <= (totalSupply() * 5) / 1000,
1215   "Swap amount cannot be higher than 0.5% total supply."
1216   );
1217   swapTokensAtAmount = newAmount;
1218
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1219

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1218   return true;
1219   }
1220
1221   function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
1222   require(
1223
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1219

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1218   return true;
1219   }
1220
1221   function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
1222   require(
1223
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1228
1229   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1230   require(
1231   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1232   "Cannot set maxWallet lower than 0.5%"
1233
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1228
1229   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1230   require(
1231   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1232   "Cannot set maxWallet lower than 0.5%"
1233
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1228
1229   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1230   require(
1231   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1232   "Cannot set maxWallet lower than 0.5%"
1233
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1231

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1230   require(
1231   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1232   "Cannot set maxWallet lower than 0.5%"
1233   );
1234   maxWallet = newNum * (10**18);
1235
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1232

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1231   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1232   "Cannot set maxWallet lower than 0.5%"
1233   );
1234   maxWallet = newNum * (10**18);
1235   }
1236
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1237

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1236
1237    function excludeFromMaxTransaction(address updAds, bool isEx)
1238    public
1239    onlyOwner
1240    {
1241
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1237

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1236
1237   function excludeFromMaxTransaction(address updAds, bool isEx)
1238   public
1239   onlyOwner
1240   {
1241
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1237

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1236
1237    function excludeFromMaxTransaction(address updAds, bool isEx)
1238    public
1239    onlyOwner
1240    {
1241
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1240   {
1241   _isExcludedMaxTransactionAmount[updAds] = isEx;
1242   }
1243
1244   // only use to disable contract sales if absolutely necessary (emergency use only)
1245
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1240   {
1241   _isExcludedMaxTransactionAmount[updAds] = isEx;
1242   }
1243
1244   // only use to disable contract sales if absolutely necessary (emergency use only)
1245
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1262

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1261   function updateSellFees(
1262   uint256 _marketingFee,
1263   uint256 _liquidityFee,
1264   uint256 _devFee
1265   ) external onlyOwner {
1266
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1262

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1261   function updateSellFees(
1262   uint256 _marketingFee,
1263   uint256 _liquidityFee,
1264   uint256 _devFee
1265   ) external onlyOwner {
1266
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1273

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1272
1273   function excludeFromFees(address account, bool excluded) public onlyOwner {
1274   _isExcludedFromFees[account] = excluded;
1275   emit ExcludeFromFees(account, excluded);
1276   }
1277
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1273

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1272
1273   function excludeFromFees(address account, bool excluded) public onlyOwner {
1274   _isExcludedFromFees[account] = excluded;
1275   emit ExcludeFromFees(account, excluded);
1276   }
1277
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1374

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1373   //when sell
1374   else if (
1375   automatedMarketMakerPairs[to] &&
1376   !_isExcludedMaxTransactionAmount[from]
1377   ) {
1378
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1391

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1390
1391   uint256 contractTokenBalance = balanceOf(address(this));
1392
1393   bool canSwap = contractTokenBalance >= swapTokensAtAmount;
1394
1395
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1420

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1419
1420    bool takeFee = !swapping;
1421
1422    // if any account belongs to _isExcludedFromFee account then remove the fee
1423    if (_isExcludedFromFees[from] || _isExcludedFromFees[to]) {
1424
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1434   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1435   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1436   }
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1434   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1435   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1436   }
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1434   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1435   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1436   }
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1437   // on buy
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1439

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1443
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1439

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1438   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1439   fees = amount.mul(buyTotalFees).div(100);
1440   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1443
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1442

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1443   }
1444
1445   if (fees > 0) {
1446
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1442

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1443   }
1444
1445   if (fees > 0) {
1446
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1442

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1441   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1442   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1443   }
1444
1445   if (fees > 0) {
1446
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1445

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1444
1445   if (fees > 0) {
1446   super._transfer(from, address(this), fees);
1447   }
1448
1449
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1446

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1445   if (fees > 0) {
1446   super._transfer(from, address(this), fees);
1447   }
1448
1449   amount -= fees;
1450
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1446

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1445   if (fees > 0) {
1446   super._transfer(from, address(this), fees);
1447   }
1448
1449   amount -= fees;
1450
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1447

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1446   super._transfer(from, address(this), fees);
1447   }
1448
1449   amount -= fees;
1450   }
1451
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1449

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1448
1449   amount -= fees;
1450   }
1451
1452   super._transfer(from, to, amount);
1453
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1449

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1448
1449   amount -= fees;
1450   }
1451
1452   super._transfer(from, to, amount);
1453
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1456

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1455   function swapTokensForEth(uint256 tokenAmount) private {
1456   // generate the uniswap pair path of token -> weth
1457   address[] memory path = new address[](2);
1458   path[0] = address(this);
1459   path[1] = uniswapV2Router.WETH();
1460
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1496

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1495   if (contractBalance == 0 || totalTokensToSwap == 0) {
1496   return;
1497   }
1498
1499   if (contractBalance > swapTokensAtAmount * 20) {
1500
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1496

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1495   if (contractBalance == 0 || totalTokensToSwap == 0) {
1496   return;
1497   }
1498
1499   if (contractBalance > swapTokensAtAmount * 20) {
1500
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1504

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1503   // Halve the amount of liquidity tokens
1504   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1505   totalTokensToSwap /
1506   2;
1507   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1508
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1505

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1504   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1505   totalTokensToSwap /
1506   2;
1507   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1508
1509
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1509

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1508
1509    uint256 initialETHBalance = address(this).balance;
1510
1511    swapTokensForEth(amountToSwapForETH);
1512
1513
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1509

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1508
1509   uint256 initialETHBalance = address(this).balance;
1510
1511   swapTokensForEth(amountToSwapForETH);
1512
1513
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1509

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1508
1509   uint256 initialETHBalance = address(this).balance;
1510
1511   swapTokensForEth(amountToSwapForETH);
1512
1513
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1526

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1525
1526   (success, ) = address(devWallet).call{value: ethForDev}("");
1527
1528   if (liquidityTokens > 0 && ethForLiquidity > 0) {
1529   addLiquidity(liquidityTokens, ethForLiquidity);
1530
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1526

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1525
1526   (success, ) = address(devWallet).call{value: ethForDev}("");
1527
1528   if (liquidityTokens > 0 && ethForLiquidity > 0) {
1529   addLiquidity(liquidityTokens, ethForLiquidity);
1530
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1592

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1591   );
1592   require(percent <= 1000, "May not nuke more than 10% of tokens in LP");
1593   lastManualLpBurnTime = block.timestamp;
1594
1595   // get balance of liquidity pair
1596
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 20

low SEVERITY

The current pragma Solidity directive is ""=0.8.10>=0.8.10>=0.8.0<0.9.0"". 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

- StreamerInu.sol

Locations

```
19    ////// lib/openzeppelin-contracts/contracts/utils/Context.sol
20    // OpenZeppelin Contracts v4.4.0 (utils/Context.sol)
21
22    /* pragma solidity ^0.8.0; */
23
24
```

SWC-115 | USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.

LINE 1354

low SEVERITY

The tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead.

Source File

- StreamerInu.sol

Locations

```
1353   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1354   );
1355   _holderLastTransferTimestamp[tx.origin] = block.number;
1356   }
1357   }
1358
```

SWC-115 | USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.

LINE 1361

low SEVERITY

Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

Source File

- StreamerInu.sol

Locations

```
1360   if (
1361   automatedMarketMakerPairs[from] &&
1362   !_isExcludedMaxTransactionAmount[to]
1363   ) {
1364   require(
1365
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1464

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1463   // make the swap
1464   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1465   tokenAmount,
1466   0, // accept any amount of ETH
1467   path,
1468
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1464

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1463   // make the swap
1464   uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
1465   tokenAmount,
1466   0, // accept any amount of ETH
1467   path,
1468
```

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

LINE 1355

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1354   );
1355   _holderLastTransferTimestamp[tx.origin] = block.number;
1356   }
1357   }
1358
1359
```

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

LINE 1362

low SEVERITY

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

Source File

- StreamerInu.sol

Locations

```
1361   automatedMarketMakerPairs[from] &&
1362   !_isExcludedMaxTransactionAmount[to]
1363   ) {
1364   require(
1365   amount <= maxTransactionAmount,
1366
```

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.