Apefund V2

Smart Contract Audit Report

SYSFIXED

09 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Apefund V2	AFND	Binance Smart Chain

Addresses

Contract address	0x7BE236a96e53dc7B1069f0212CB77d33aD8CBacF
Contract deployer address	0x04dd4D558D54013c1c8C6AaA0e2F9F66017c2A40

Project Website

https://apefunddao.com/

Codebase

https://bscscan.com/address/0x7BE236a96e53dc7B1069f0212CB77d33aD8CBacF#code

SUMMARY

The first "Social investment community" token. A private community for the true bsc apes. Get access to the community only with $afnd tokens. Community driven vested interest unique insight out advantage earn & invest as a community, connect, discuss & learn, consensus voting, ongoing review of investments tokenomics: 5% buys 15% sells.

Contract Summary

Documentation Quality

Apefund V2 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 Apefund V2 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 105, 180 and 192.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 119, 119, 187, 187, 188, 188, 317, 345, 385, 385, 407, 419, 419, 419, 420, 421, 434, 434, 434, 434, 435, 435, 439, 439, 439, 440, 440, 444, 444, 448, 448, 452, 452, 456, 456, 457, 457, 459, 459, 460, 461, 537, 551, 551, 571, 571, 571, 572, 589, 589, 622, 623, 623, 624, 624, 625, 625, 660, 660, 661, 661, 678, 679, 679, 680, 680, 694, 696, 709, 722, 722, 723, 723, 724, 726, 730 and 734.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- 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 575, 576, 607, 608, 679, 680 and 680.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 498.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 657.

CONCLUSION

We have audited the NamaProject Coin which has released on January 2023 to discover issues and identify potential security vulnerabilities in NamaProject 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 provides a satisfactory result with some low-risk issues.

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, out of bounds array access, and potential use of "block.number" as a source of randomness. We recommend to Don't using any of those environment variables as sources of randomness and being aware that the use of these variables introduces a certain level of trust in miners, 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, and 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.

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 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	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.	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
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
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	Sunday Jan 08 2023 15:36:27 GMT+0000 (Coordinated Universal Time)
Finished	Monday Jan 09 2023 13:48:21 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	ApefundV2.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-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-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-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-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 119

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
118   uint8 constant private _decimals = 18;
119   uint256 constant private _tTotal = startingSupply * 10**_decimals;
120
121   struct Fees {
122   uint16 buyFee;
123
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 119

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
118   uint8 constant private _decimals = 18;
119   uint256 constant private _tTotal = startingSupply * 10**_decimals;
120
121   struct Fees {
122   uint16 buyFee;
123
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 187

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
186
187   uint256 private _maxTxAmount = (_tTotal * 3) / 100;
188   uint256 private _maxWalletSize = (_tTotal * 3) / 100;
189
190   bool public tradingEnabled = false;
191
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 187

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
186
187   uint256 private _maxTxAmount = (_tTotal * 3) / 100;
188   uint256 private _maxWalletSize = (_tTotal * 3) / 100;
189
190   bool public tradingEnabled = false;
191
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 188

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
187   uint256 private _maxTxAmount = (_tTotal * 3) / 100;
188   uint256 private _maxWalletSize = (_tTotal * 3) / 100;
189
190   bool public tradingEnabled = false;
191   bool public _hasLiqBeenAdded = false;
192
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 188

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
187   uint256 private _maxTxAmount = (_tTotal * 3) / 100;
188   uint256 private _maxWalletSize = (_tTotal * 3) / 100;
189
190   bool public tradingEnabled = false;
191   bool public _hasLiqBeenAdded = false;
192
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 317

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
316   if (_allowances[sender][msg.sender] != type(uint256).max) {
317   _allowances[sender][msg.sender] -= amount;
318   }
319
320   return _transfer(sender, recipient, amount);
321
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 345

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
344   if (timeSinceLastPair != 0) {
345   require(block.timestamp - timeSinceLastPair > 3 days, "3 Day cooldown.");
346   }
347   require(!lpPairs[pair], "Pair already added to list.");
348   lpPairs[pair] = true;
349
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 385

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
384   function getCirculatingSupply() public view returns (uint256) {
385   return (_tTotal - (balanceOf(DEAD) + balanceOf(address(0))));
386   }
387
388   function removeSniper(address account) external onlyOwner {
389
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 385

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
384   function getCirculatingSupply() public view returns (uint256) {
385   return (_tTotal - (balanceOf(DEAD) + balanceOf(address(0))));
386   }
387
388   function removeSniper(address account) external onlyOwner {
389
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
406   "Cannot exceed maximums.");
407   require(buyFee + sellFee <= maxRoundtripTax, "Cannot exceed roundtrip maximum.");
408   _taxRates.buyFee = buyFee;
409   _taxRates.sellFee = sellFee;
410   _taxRates.transferFee = transferFee;
411
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 419

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
418   _ratios.rewards = rewards;
419   _ratios.totalSwap = liquidity + investment + futureTax + opex;
420   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
421   require(_ratios.totalSwap + _ratios.rewards <= total, "Cannot exceed sum of buy and
sell fees.");
422   }
423
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 419

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
418   _ratios.rewards = rewards;
419   _ratios.totalSwap = liquidity + investment + futureTax + opex;
420   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
421   require(_ratios.totalSwap + _ratios.rewards <= total, "Cannot exceed sum of buy and
sell fees.");
422   }
423
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 419

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
418   _ratios.rewards = rewards;
419   _ratios.totalSwap = liquidity + investment + futureTax + opex;
420   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
421   require(_ratios.totalSwap + _ratios.rewards <= total, "Cannot exceed sum of buy and
sell fees.");
422   }
423
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 420

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
419   _ratios.totalSwap = liquidity + investment + futureTax + opex;
420   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
421   require(_ratios.totalSwap + _ratios.rewards <= total, "Cannot exceed sum of buy and
sell fees.");
422   }
423
424
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 421

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
420   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
421   require(_ratios.totalSwap + _ratios.rewards <= total, "Cannot exceed sum of buy and
sell fees.");
422   }
423
424   function setWallets(address payable rewards, address payable investment, address
payable opex, address payable liquidity, address payable futureTax) external onlyOwner {
425
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
433   function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
433   function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
433   function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
433   function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438   function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
439
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
434   require((_tTotal * percent) / divisor >= (_tTotal * 5 / 1000), "Max Transaction amt
must be above 0.5% of total supply.");
435   _maxTxAmount = (_tTotal * percent) / divisor;
436   }
437
438   function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
439
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
438   function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
439   require((_tTotal * percent) / divisor >= (_tTotal / 100), "Max Wallet amt must be
above 1% of total supply.");
440   _maxWalletSize = (_tTotal * percent) / divisor;
441   }
442
443
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
438   function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
439   require((_tTotal * percent) / divisor >= (_tTotal / 100), "Max Wallet amt must be
above 1% of total supply.");
440   _maxWalletSize = (_tTotal * percent) / divisor;
441   }
442
443
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
438   function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
439   require((_tTotal * percent) / divisor >= (_tTotal / 100), "Max Wallet amt must be
above 1% of total supply.");
440   _maxWalletSize = (_tTotal * percent) / divisor;
441   }
442
443
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 440

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
439   require((_tTotal * percent) / divisor >= (_tTotal / 100), "Max Wallet amt must be
above 1% of total supply.");
440   _maxWalletSize = (_tTotal * percent) / divisor;
441   }
442
443   function getMaxTX() external view returns (uint256) {
444
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 440

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
439   require((_tTotal * percent) / divisor >= (_tTotal / 100), "Max Wallet amt must be
above 1% of total supply.");
440   _maxWalletSize = (_tTotal * percent) / divisor;
441   }
442
443   function getMaxTX() external view returns (uint256) {
444
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 444

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
443   function getMaxTX() external view returns (uint256) {
444   return _maxTxAmount / (10**_decimals);
445   }
446
447   function getMaxWallet() external view returns (uint256) {
448
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 444

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
443   function getMaxTX() external view returns (uint256) {
444   return _maxTxAmount / (10**_decimals);
445   }
446
447   function getMaxWallet() external view returns (uint256) {
448
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 448

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
447   function getMaxWallet() external view returns (uint256) {
448   return _maxWalletSize / (10**_decimals);
449   }
450
451   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
452
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 448

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
447   function getMaxWallet() external view returns (uint256) {
448   return _maxWalletSize / (10**_decimals);
449   }
450
451   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
452
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 452

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
451   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
452   return((balanceOf(lpPair) * priceImpactInHundreds) / masterTaxDivisor);
453   }
454
455   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
456
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 452

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
451   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
452   return((balanceOf(lpPair) * priceImpactInHundreds) / masterTaxDivisor);
453   }
454
455   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
456
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 456

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
455   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
456   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
457   swapAmount = (_tTotal * amountPercent) / amountDivisor;
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 456

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
455   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
456   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
457   swapAmount = (_tTotal * amountPercent) / amountDivisor;
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 457

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
456   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
457   swapAmount = (_tTotal * amountPercent) / amountDivisor;
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 457

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
456   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
457   swapAmount = (_tTotal * amountPercent) / amountDivisor;
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 459

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
462   }
463
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 459

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
458   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
462   }
463
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 460

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
459   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
462   }
463
464
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 461

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
460   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
461   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
462   }
463
464   function setPriceImpactSwapAmount(uint256 priceImpactSwapPercent) external
onlyOwner {
465
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 537

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
536   if (!_isExcludedFromLimits[to]) {
537   require(balanceOf(to) + amount <= _maxWalletSize, "Transfer amount exceeds the
maxWalletSize.");
538   }
539   }
540   }
541
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 551

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
550   uint256 swapAmt = swapAmount;
551   if (piContractSwapsEnabled) { swapAmt = (balanceOf(lpPair) * piSwapPercent) /
masterTaxDivisor; }
552   if (contractTokenBalance >= swapAmt) { contractTokenBalance = swapAmt; }
553   contractSwap(contractTokenBalance);
554   }
555
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 551

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
550   uint256 swapAmt = swapAmount;
551   if (piContractSwapsEnabled) { swapAmt = (balanceOf(lpPair) * piSwapPercent) /
masterTaxDivisor; }
552   if (contractTokenBalance >= swapAmt) { contractTokenBalance = swapAmt; }
553   contractSwap(contractTokenBalance);
554   }
555
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 571

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
570
571   uint256 toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap)
/ 2;
572   uint256 swapAmt = contractTokenBalance - toLiquify;
573
574   address[] memory path = new address[](2);
575
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 571

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
570
571   uint256 toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap)
/ 2;
572   uint256 swapAmt = contractTokenBalance - toLiquify;
573
574   address[] memory path = new address[](2);
575
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 571

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
570
571   uint256 toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap)
/ 2;
572   uint256 swapAmt = contractTokenBalance - toLiquify;
573
574   address[] memory path = new address[](2);
575
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
571   uint256 toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap)
/ 2;
572   uint256 swapAmt = contractTokenBalance - toLiquify;
573
574   address[] memory path = new address[](2);
575   path[0] = address(this);
576
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 589

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
588   uint256 amtBalance = address(this).balance;
589   uint256 liquidityBalance = (amtBalance * toLiquify) / swapAmt;
590
591   if (toLiquify > 0) {
592   try dexRouter.addLiquidityETH{value: liquidityBalance}(
593
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 589

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
588   uint256 amtBalance = address(this).balance;
589   uint256 liquidityBalance = (amtBalance * toLiquify) / swapAmt;
590
591   if (toLiquify > 0) {
592   try dexRouter.addLiquidityETH{value: liquidityBalance}(
593
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 622

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
621   amtBalance = IERC20_BUSD.balanceOf(address(this));
622   ratios.totalSwap -= ratios.liquidity;
623   uint256 investmentBalance = (amtBalance * ratios.investment) / ratios.totalSwap;
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
622   ratios.totalSwap -= ratios.liquidity;
623   uint256 investmentBalance = (amtBalance * ratios.investment) / ratios.totalSwap;
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
622   ratios.totalSwap -= ratios.liquidity;
623   uint256 investmentBalance = (amtBalance * ratios.investment) / ratios.totalSwap;
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 624

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
623   uint256 investmentBalance = (amtBalance * ratios.investment) / ratios.totalSwap;
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627   IERC20_BUSD.transfer(_taxWallets.opex, opexBalance);
628
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 624

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
623   uint256 investmentBalance = (amtBalance * ratios.investment) / ratios.totalSwap;
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627   IERC20_BUSD.transfer(_taxWallets.opex, opexBalance);
628
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 625

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627   IERC20_BUSD.transfer(_taxWallets.opex, opexBalance);
628   }
629
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 625

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
624   uint256 futureTaxBalance = (amtBalance * ratios.futureTax) / ratios.totalSwap;
625   uint256 opexBalance = amtBalance - (investmentBalance + futureTaxBalance);
626   if (ratios.opex > 0) {
627   IERC20_BUSD.transfer(_taxWallets.opex, opexBalance);
628   }
629
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 660

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
659   allowedPresaleExclusion = false;
660   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
661   swapAmount = (balanceOf(lpPair) * 30) / 10000;
662   launchStamp = block.timestamp;
663   }
664
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 660

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
659   allowedPresaleExclusion = false;
660   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
661   swapAmount = (balanceOf(lpPair) * 30) / 10000;
662   launchStamp = block.timestamp;
663   }
664
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 661

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
660   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
661   swapAmount = (balanceOf(lpPair) * 30) / 10000;
662   launchStamp = block.timestamp;
663   }
664
665
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 661

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
660   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
661   swapAmount = (balanceOf(lpPair) * 30) / 10000;
662   launchStamp = block.timestamp;
663   }
664
665
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 678

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
677   require(accounts.length == amounts.length, "Lengths do not match.");
678   for (uint16 i = 0; i < accounts.length; i++) {
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 679

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
678   for (uint16 i = 0; i < accounts.length; i++) {
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 679

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
678   for (uint16 i = 0; i < accounts.length; i++) {
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 680

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
684
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 680

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
684
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 694

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
693   }
694   _tOwned[from] -= amount;
695   uint256 amountReceived = (takeFee) ? takeTaxes(from, buy, sell, amount) : amount;
696   _tOwned[to] += amountReceived;
697   emit Transfer(from, to, amountReceived);
698
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 696

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
695   uint256 amountReceived = (takeFee) ? takeTaxes(from, buy, sell, amount) : amount;
696   _tOwned[to] += amountReceived;
697   emit Transfer(from, to, amountReceived);
698   if (!_hasLiqBeenAdded) {
699   _checkLiquidityAdd(from, to);
700
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 709

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
708   Ratios memory ratios = _ratios;
709   uint256 total = ratios.totalSwap + ratios.rewards;
710   uint256 currentFee;
711   if (buy) {
712   currentFee = _taxRates.buyFee;
713
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 722

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
721   || block.chainid == 56)) { currentFee = 4500; }
722   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
723   uint256 rewardsAmount = feeAmount * ratios.rewards / total;
724   uint256 swapAmt = feeAmount - rewardsAmount;
725   if (swapAmt > 0) {
726
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 722

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
721   || block.chainid == 56)) { currentFee = 4500; }
722   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
723   uint256 rewardsAmount = feeAmount * ratios.rewards / total;
724   uint256 swapAmt = feeAmount - rewardsAmount;
725   if (swapAmt > 0) {
726
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
722   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
723   uint256 rewardsAmount = feeAmount * ratios.rewards / total;
724   uint256 swapAmt = feeAmount - rewardsAmount;
725   if (swapAmt > 0) {
726   _tOwned[address(this)] += swapAmt;
727
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
722   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
723   uint256 rewardsAmount = feeAmount * ratios.rewards / total;
724   uint256 swapAmt = feeAmount - rewardsAmount;
725   if (swapAmt > 0) {
726   _tOwned[address(this)] += swapAmt;
727
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 724

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
723   uint256 rewardsAmount = feeAmount * ratios.rewards / total;
724   uint256 swapAmt = feeAmount - rewardsAmount;
725   if (swapAmt > 0) {
726   _tOwned[address(this)] += swapAmt;
727   emit Transfer(from, address(this), swapAmt);
728
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 726

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
725   if (swapAmt > 0) {
726   _tOwned[address(this)] += swapAmt;
727   emit Transfer(from, address(this), swapAmt);
728   }
729   if (rewardsAmount > 0) {
730
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 730

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
729   if (rewardsAmount > 0) {
730   _tOwned[_taxWallets.rewards] += rewardsAmount;
731   emit Transfer(from, _taxWallets.rewards, rewardsAmount);
732   }
733
734
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 734

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
733
734   return amount - feeAmount;
735   }
736   }
737
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is ""=0.6.0<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

- ApefundV2.sol

Locations

```
5   // SPDX-License-Identifier: MIT
6   pragma solidity >=0.6.0 <0.9.0;
7
8   interface IERC20 {
9   function totalSupply() external view returns (uint256);
10
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 105

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
104   mapping (address => uint256) private _tOwned;
105   mapping (address => bool) lpPairs;
106   uint256 private timeSinceLastPair = 0;
107   mapping (address => mapping (address => uint256)) private _allowances;
108   mapping (address => bool) private _liquidityHolders;
109
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 180

low 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

- ApefundV2.sol

Locations

```
179
180   bool inSwap;
181   bool public contractSwapEnabled = false;
182   uint256 public swapThreshold;
183   uint256 public swapAmount;
184
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 192

low SEVERITY

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.

Source File

- ApefundV2.sol

Locations

```
191   bool public _hasLiqBeenAdded = false;
192   Protections protections;
193   uint256 public launchStamp;
194
195   event ContractSwapEnabledUpdated(bool enabled);
196
```

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

LINE 498

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

- ApefundV2.sol

Locations

```
497   && to != _owner
498   && tx.origin != _owner
499   && !_liquidityHolders[to]
500   && !_liquidityHolders[from]
501   && to != DEAD
502
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 575

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
574   address[] memory path = new address[](2);
575   path[0] = address(this);
576   path[1] = dexRouter.WETH();
577
578   try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
579
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 576

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
575   path[0] = address(this);
576   path[1] = dexRouter.WETH();
577
578   try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
579   swapAmt,
580
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 607

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
606   path = new address[](2);
607   path[0] = dexRouter.WETH();
608   path[1] = BUSD;
609
610   try dexRouter.swapExactETHForTokens{value: address(this).balance}
611
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 608

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
607   path[0] = dexRouter.WETH();
608   path[1] = BUSD;
609
610   try dexRouter.swapExactETHForTokens{value: address(this).balance}
611   (
612
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 679

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
678   for (uint16 i = 0; i < accounts.length; i++) {
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 680

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
684
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 680

low SEVERITY

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

Source File

- ApefundV2.sol

Locations

```
679   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
680   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
681   }
682   }
683
684
```

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

LINE 657

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

- ApefundV2.sol

Locations

```
656   }
657   try protections.setLaunch(lpPair, uint32(block.number), uint64(block.timestamp),
_decimals) {} catch {}
658   tradingEnabled = true;
659   allowedPresaleExclusion = false;
660   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
661
```

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.