

Started	Tue Jul 19 2022 08:30:26 GMT+0000 (Coordinated Universal Time)
Finished	Tue Jul 19 2022 08:32:15 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Vscode-Extension
Main Source File	/Contracts/Babyaapt.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	1

ISSUES

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
87 dividendTracker = new BABYAAPTDividendTracker();
88
89 IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
90 0x10ED43C718714eb63d5aA57B78854704E256024E
91 );
```

UNKNOWN Arithmetic operation "***" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
87 dividendTracker = new BABYAAPTDividendTracker();
88
89 IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
90 0x10ED43C718714eb63d5aA57B78854704E256024E
91 );
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
147 |  
148 | function updateUniswapV2Router(address newAddress) public onlyOwner {  
149 | require(newAddress != address(uniswapV2Router), "Same");  
150 | uniswapV2Router = IUniswapV2Router02(newAddress);  
151 | address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router).factory()
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
147 |  
148 | function updateUniswapV2Router(address newAddress) public onlyOwner {  
149 | require(newAddress != address(uniswapV2Router), "Same");  
150 | uniswapV2Router = IUniswapV2Router02(newAddress);  
151 | address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router).factory()
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
390 | _buyBackWalletAddress,  
391 | newBalance.mul(buyBackFee).div(marketingFee.add(buyBackFee))  
392 | );  
393 |  
394 | if marketingFee > 0 {  
395 | IERC20(BUSD).transfer(  
396 | _marketingWalletAddress,
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
543 | returns (  
544 | address account,  
545 | int256 index,  
546 | int256 iterationsUntilProcessed,  
547 | uint256 withdrawableDividends,  
548 | uint256 totalDividends,
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/babyaapt.sol

Locations

```
544 | address account,  
545 | int256 index,  
546 | int256 iterationsUntilProcessed,  
547 | uint256 withdrawableDividends,  
548 | uint256 totalDividends,
```

LOW State variable visibility is not set.

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

SWC-108

Source file

/contracts/babyaapt.sol

Locations

```
56 | address(0x2a6f8E3a8Fe92849B08CA82F4f323DD43029B33D); //AAPT  
57 |  
58 | bool isRewardBUSD; // true: reward as BUSD, false: reward as AAPT  
59 | uint256 public swapTokensAtAmount = 2000000 * (10**18);  
60 |
```

UNKNOWN Out of bounds array access

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

SWC-110

Source file

/contracts/babyaapt.sol

Locations

```
454 | function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
455 | // approve token transfer to cover all possible scenarios
456 | _approve(address(this), address(uniswapV2Router), tokenAmount);
457 |
458 | // add the liquidity
```

UNKNOWN Out of bounds array access

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

SWC-110

Source file

/contracts/babyaapt.sol

Locations

```
456 | _approve(address(this), address(uniswapV2Router), tokenAmount);
457 |
458 | // add the liquidity
459 | uniswapV2Router.addLiquidityETH(value: ethAmount){
460 | address(this),
```

UNKNOWN Out of bounds array access

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

SWC-110

Source file

/contracts/babyaapt.sol

Locations

```
470 | swapTokensForBUSD(tokens, isRewardBUSD);
471 | address processingToken = isRewardBUSD ? BUSD : AAPT;
472 | uint256 dividends = IERC20(processingToken).balanceOf(address(this));
473 | bool success = IERC20(processingToken).transfer(
474 | address(dividendTracker),
```

UNKNOWN Out of bounds array access

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

SWC-110

Source file

/contracts/babyaapt.sol

Locations

```
471 | address processingToken = isRewardBUSD ? BUSD : AAPT;  
472 | uint256 dividends = IERC20(processingToken).balanceOf(address(this));  
473 | bool success = IERC20(processingToken).transfer(  
474 | address(dividendTracker),  
475 | dividends
```

UNKNOWN Out of bounds array access

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

SWC-110

Source file

/contracts/babyaapt.sol

Locations

```
471 | address processingToken = isRewardBUSD ? BUSD : AAPT;  
472 | uint256 dividends = IERC20(processingToken).balanceOf(address(this));  
473 | bool success = IERC20(processingToken).transfer(  
474 | address(dividendTracker),  
475 | dividends
```