

Started	Sun Mar 05 2023 19:44:07 GMT+0000 (Coordinated Universal Time)
Finished	Sun Mar 05 2023 19:44:44 GMT+0000 (Coordinated Universal Time)
Mode	Quick
Client Tool	Mythx-Cli-0.7.3
Main Source File	Contracts/MyOwnTokenBonding-Flat.sol

## DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	13

## ISSUES

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
408 | function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
409 |     address owner = _msgSender();
410 |     _approve(owner, spender, allowance[owner][spender] + addedValue);
411 |     return true;
412 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
431 | require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below zero");
432 | unchecked {
433 |     _approve(owner, spender, currentAllowance - subtractedValue);
434 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
464 | require(fromBalance >= amount, "ERC20: transfer amount exceeds balance");
465 | unchecked {
466 |   _balances[from] = fromBalance - amount;
467 |   // Overflow not possible: the sum of all balances is capped by totalSupply, and the sum is preserved by
468 |   // decrementing then incrementing.
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
467 | // Overflow not possible: the sum of all balances is capped by totalSupply, and the sum is preserved by
468 | // decrementing then incrementing.
469 | _balances[to] += amount;
470 | }
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
489 | _beforeTokenTransfer(address(0), account, amount);
490 |
491 | totalSupply += amount;
492 | unchecked {
493 |   // Overflow not possible: balance + amount is at most totalSupply + amount, which is checked above.
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
492 | unchecked {  
493 | // Overflow not possible: balance + amount is at most totalSupply + amount, which is checked above.  
494 | _balances[account] += amount;  
495 | }  
496 | emit Transfer(address(0), account, amount);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
518 | require(accountBalance >= amount, "ERC20: burn amount exceeds balance");  
519 | unchecked {  
520 | _balances[account] = accountBalance - amount;  
521 | // Overflow not possible: amount <= accountBalance <= totalSupply.  
522 | _totalSupply -= amount;
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
520 | _balances[account] = accountBalance - amount;  
521 | // Overflow not possible: amount <= accountBalance <= totalSupply.  
522 | _totalSupply -= amount;  
523 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
570 | require(currentAllowance >= amount, "ERC20: insufficient allowance");
571 | unchecked {
572 |   _approve(owner, spender, currentAllowance - amount);
573 | }
574 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
649 | */
650 | function _mint(address account, uint256 amount) internal virtual override {
651 |   require(ERC20.totalSupply() + amount <= cap(), "ERC20Capped: cap exceeded");
652 |   super._mint(account, amount);
653 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1456 | ) external returns (bytes4) {
1457 |   require(from != address(0), "Minting to smart contract fails");
1458 |   uint256 _currentPrice = BASIC_PRICE +
1459 |     PRICE_PER_TOKEN * totalSupply() /
1460 |     10 ** decimals();
1461 |   uint256 curveBasePrice = ((value * _currentPrice) / 10 ** decimals());
1462 |   uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1457 | require(from != address(0), "Minting to smart contract fails");
1458 | uint256 _currentPrice = BASIC_PRICE +
1459 |     PRICE_PER_TOKEN * totalSupply() /
1460 |     10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1457 | require(from != address(0), "Minting to smart contract fails");
1458 | uint256 _currentPrice = BASIC_PRICE +
1459 |     (PRICE_PER_TOKEN * totalSupply()) /
1460 |     10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1458 | uint256 _currentPrice = BASIC_PRICE +
1459 |     (PRICE_PER_TOKEN * totalSupply()) /
1460 |     10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1459 | (PRICE_PER_TOKEN * totalSupply()) /  
1460 | 10 ** decimals();  
1461 | uint256 curveBasePrice = ((value * _currentPrice) / 10 ** decimals());  
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /  
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1459 | (PRICE_PER_TOKEN * totalSupply()) /  
1460 | 10 ** decimals();  
1461 | uint256 curveBasePrice = ((value * _currentPrice) / 10 ** decimals());  
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /  
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1459 | (PRICE_PER_TOKEN * totalSupply()) /  
1460 | 10 ** decimals();  
1461 | uint256 curveBasePrice = ((value * _currentPrice) / 10 ** decimals());  
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /  
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1460 | 10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = ((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1460 | 10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = ((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1460 | 10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = ((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1460 | 10 ** decimals();
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```



## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1461 | uint256 curveBasePrice = ((value * _currentPrice)) / 10 ** decimals();
1462 | uint256 curveExtraPrice = (((value * PRICE_PER_TOKEN) /
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1463 | 10 ** decimals()) * (value)) / (2 * 10 ** decimals());
1464 | _burn(address(this), value);
1465 | payable(from).transfer(curveBasePrice - curveExtraPrice);
1466 | return
1467 | bytes4(
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1476 | */
1477 | function buyPriceCalculation(uint256 amount) public view returns (uint256) {
1478 | uint256 _currentPrice = BASIC_PRICE +
1479 | PRICE_PER_TOKEN * totalSupply() /
1480 | 10 ** decimals();
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1477 | function buyPriceCalculation(uint256 amount) public view returns (uint256) {
1478 |     uint256 _currentPrice = BASIC_PRICE +
1479 |     (PRICE_PER_TOKEN * totalSupply()) /
1480 |     10 ** decimals();
1481 |     uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());
1482 |     uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1477 | function buyPriceCalculation(uint256 amount) public view returns (uint256) {
1478 |     uint256 _currentPrice = BASIC_PRICE +
1479 |     (PRICE_PER_TOKEN * totalSupply()) /
1480 |     10 ** decimals();
1481 |     uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1478 |     uint256 _currentPrice = BASIC_PRICE +
1479 |     (PRICE_PER_TOKEN * totalSupply()) /
1480 |     10 ** decimals();
1481 |     uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());
1482 |     uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1479 | (PRICE_PER_TOKEN * totalSupply()) /  
1480 | 10 ** decimals();  
1481 | uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());  
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /  
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1479 | (PRICE_PER_TOKEN * totalSupply()) /  
1480 | 10 ** decimals();  
1481 | uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());  
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /  
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1479 | (PRICE_PER_TOKEN * totalSupply()) /  
1480 | 10 ** decimals();  
1481 | uint256 curveBasePrice = ((amount * _currentPrice) / 10 ** decimals());  
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /  
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1480 | 10 ** decimals();
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = ((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1480 | 10 ** decimals();
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = ((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1480 | 10 ** decimals();
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = ((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1480 | 10 ** decimals();
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1481 | uint256 curveBasePrice = ((amount * _currentPrice)) / 10 ** decimals();
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1482 | uint256 curveExtraPrice = (((amount * PRICE_PER_TOKEN) /
1483 | 10 ** decimals()) * amount) / (2 * 10 ** decimals());
1484 | return (curveBasePrice + curveExtraPrice);
1485 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1489 | */
1490 | function currentPrice() external view returns (uint256) {
1491 | return PRICE_PER_TOKEN * totalSupply;
1492 | }
1493 | }
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
6  // OpenZeppelin Contracts v4.4.1 (utils/Context.sol)
7
8  pragma solidity ^0.8.0;
9
10 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
33 // OpenZeppelin Contracts (last updated v4.7.0) (access/Ownable.sol)
34
35 pragma solidity ^0.8.0;
36
37 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
117 // OpenZeppelin Contracts (last updated v4.6.0) (token/ERC20/IERC20.sol)
118
119 pragma solidity ^0.8.0;
120
121 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
202 // OpenZeppelin Contracts v4.4.1 (token/ERC20/extensions/IERC20Metadata.sol)
203
204 pragma solidity ^0.8.0;
205
206 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
231 // OpenZeppelin Contracts (last updated v4.8.0) (token/ERC20/ERC20.sol)
232
233 pragma solidity ^0.8.0;
234
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
621 // OpenZeppelin Contracts v4.4.1 (token/ERC20/extensions/ERC20Capped.sol)
622
623 pragma solidity ^0.8.0;
624
625 /**
```



LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
659 // OpenZeppelin Contracts v4.4.1 (utils/introspection/IERC165.sol)
660
661 pragma solidity ^0.8.0;
662
663 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
687 // OpenZeppelin Contracts v4.4.1 (utils/introspection/ERC165.sol)
688
689 pragma solidity ^0.8.0;
690
691 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.1"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
717 // OpenZeppelin Contracts (last updated v4.8.0) (utils/Address.sol)
718
719 pragma solidity ^0.8.1;
720
721 /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
963 |  
964 |  
965 | pragma solidity ^0.8.0;  
966 |
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1048 |  
1049 |  
1050 | pragma solidity ^0.8.0;  
1051 |  
1052 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1086 |  
1087 |  
1088 | pragma solidity ^0.8.0;  
1089 |  
1090 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

contracts/MyOwnTokenBonding-flat.sol

Locations

```
1118 |  
1119 |  
1120 | pragma solidity ^0.8.0;  
1121 |
```