MythX

REPORT 616AC01DA995530019023FB6

Created	Sat Oct 16 2021 12:05:49 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	614184718bfa12ce53f29d58

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
0f87b398-1550-4421-a53f-4e56b1c58330	/flattenedcontracts/gausscrowdsale.sol	0

Analysis 0f87b398-1550-4421-a53f-4e56b1c58330 MythX

Started	Sat Oct 16 2021 12:05:50 GMT+0000 (Coordinated Universal Time)
Finished	Sat Oct 16 2021 12:05:55 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Vscode-Extension
Main Source File	/Flattenedcontracts/Gausscrowdsale.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	0

ISSUES

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
254 _state = State.Refunding;
255
256 for (uint i = 0; i < buyers.length; i++) {
257 require(buyers[i].amount > 0, "RefundVault: beneficiary amount can not be 0.");
258 _refund(buyers[i]);
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
387 __Ownable_init();
388 startTime = _startTime;
389 endTime = startTime + (30 days);
390 crowdsaleWallet = _crowdsaleWallet;
391 _token = IBEP20(_gaussAddress);
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
391 _token = IBEP20(_gaussAddress);
392 refundVault = new RefundVault(crowdsaleWallet);
393 minimumCap = (550 * 10**8);
394 purchaseCap = (100 * 10**8);
395 jagerRaised = 0;
```

UNKNOWN Arithmetic operation "**" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
391 _token = IBEP20(_gaussAddress);
392 refundVault = new RefundVault(crowdsaleWallet);
393 minimumCap = (550 * 10**8);
394 purchaseCap = (100 * 10**8);
395 jagerRaised = 0;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
392 refundVault = new RefundVault(crowdsaleWallet);
393 minimumCap = (550 * 10**8);
394 purchaseCap = (100 * 10**8);
395 jagerRaised = 0;
396 gaussSold = 0;
```

UNKNOWN Arithmetic operation "**" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
392 refundVault = new RefundVault(crowdsaleWallet);
393 minimumCap = (550 * 10**8);
394 purchaseCap = (100 * 10**8);
395 jagerRaised = 0;
396 gaussSold = 0;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
425 require(_jagerAmount != 0, "GaussCrowdsale: amount of BNB must be greater than 0.");
426 require(_jagerAmount <= purchaseCap, "Crowdsale: amount of BNB sent must lower than 100");
427 require((balances[_beneficiary] + _jagerAmount) <= purchaseCap, "Crowdsale: amount of BNB entered exceeds buyers purchase cap.");
428 }
429
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
433
434 // Calculates the token amount using the "jagerAmount" and the rate at the current stage.
435 uint256 tokenAmount = ((_jagerAmount * rates[currentStage])/(10**8));
436
437 // Addes the "tokenAmount" to the beneficiary's balance.
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
433
434 // Calculates the token amount using the "jagerAmount" and the rate at the current stage.
435 uint256 tokenAmount = ((_jagerAmount * rates[currentStage])/(10**8));
436
437 // Addes the "tokenAmount" to the beneficiary's balance.
```

UNKNOWN Arithmetic operation "**" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
433
434 // Calculates the token amount using the "jagerAmount" and the rate at the current stage.
435 uint256 tokenAmount = ((_jagerAmount * rates[currentStage])/(10**8));
436
437 // Addes the "tokenAmount" to the beneficiary's balance.
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
436
437 // Addes the "tokenAmount" to the beneficiary's balance.
438 balances[_beneficiary] = balances[_beneficiary] + tokenAmount;
439
440 _updatePurchasingState(tokenAmount, _jagerAmount);
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
445 // Updates the amount of tokens left in the Crowdsale; may change the stage if conditions are met.
446 function _updatePurchasingState(uint256 _tokenAmount, uint256 _jagerAmount) internal {
447 gaussSold = gaussSold + _tokenAmount;
448 jagerRaised = jagerRaised + _jagerAmount;
449
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
446 function _updatePurchasingState(uint256 _tokenAmount, uint256 _jagerAmount) internal {
447 gaussSold = gaussSold + _tokenAmount;
448 jagerRaised = jagerRaised + _jagerAmount;
449
450 if (gaussSold >= stages[currentStage]) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
450 if (gaussSold >= stages[currentStage]) {
451 if (currentStage < stages.length) {
452 currentStage = currentStage + 1;
453 }
454 }
```

UNKNOWN

SWC-110

Public state variable with array type causing reacheable exception by default.

The public state variable "buyers" in "RefundVault" contract has type "struct RefundVault.Buyer[]" and can cause an exception in case of use of invalid array index value.

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
208
209 // Array of all Buyers, used to keep track of each Buyer's address and amount of BNB spent.
210 Buyer[] public buyers;
211
212 // Address where BNB funds are collected.
```

UNKNOWN

SWC-110

Out of bounds array access

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

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
255
256 for (uint i = 0; i < buyers.length; i++) {
257 require(buyers[i].amount > 0, "RefundVault: beneficiary amount can not be 0.");
258 _refund(buyers[i]);
259 }
```

UNKNOWN

SWC-110

Out of bounds array access

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

Source file

/flattenedcontracts/gausscrowdsale.sol

Locations

```
256 for (uint i = 0; i < buyers.length; i++) {
257 require(buyers[i].amount > 0, "RefundVault: beneficiary amount can not be 0.");
258 _refund(buyers[i]);
259 }
260
```

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

/flattenedcontracts/gausscrowdsale.sol

Locations

```
433
434 // Calculates the token amount using the "jagerAmount" and the rate at the current stage.
435 uint256 tokenAmount = ((_jagerAmount * rates[currentStage])/(10**8));
436
437 // Addes the "tokenAmount" to the beneficiary's balance.
```

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

/flattenedcontracts/gausscrowdsale.sol

Locations

```
448 jagerRaised = jagerRaised + _jagerAmount;
449
450 if (gaussSold >= stages[currentStage]) {
451 if (currentStage < stages.length) {
452 currentStage = currentStage + 1;
```