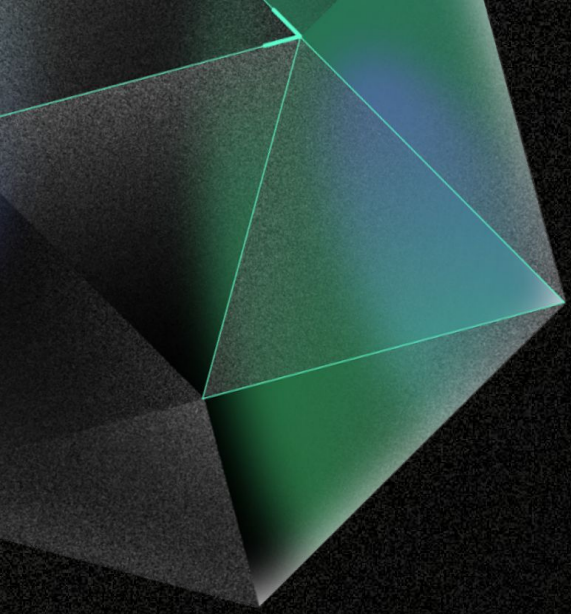
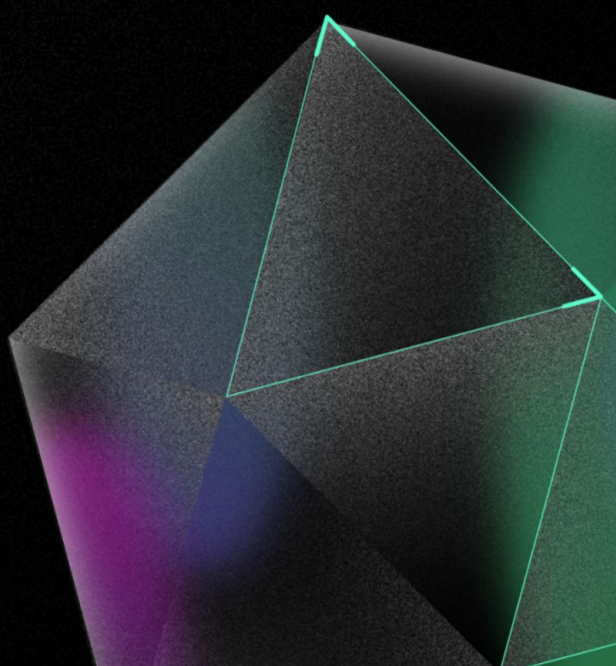


hexens

SMART CONTRACT AUDIT REPORT FOR EUPHORIA.GAMES

01.12.2021



Contents

Summary

Euphoria's smart contracts have low risk severity. During the audit our specialists found multiple low risk weaknesses and one medium risk weakness.

Severity	Number of Findings
High	0
Medium	1
Low	11
Informational	0

Total: 12

Scope

The analyzed contracts are located in the following repository:
https://github.com/EuphoriaHit/Contracts/blob/main/contracts/Staking.sol
https://github.com/EuphoriaHit/Contracts/blob/contracts/BridgeBsc.sol
https://github.com/EuphoriaHit/Contracts/blob/contracts/BridgeEth.sol
(commit 5cc2a75d22bb9fb2d9fc297ff73bba7cda8392d8)

Description

This audit covered the Staking, BridgeBsc and BridgeEth contracts of Euphoria.

Test Cases

Testing results are presented below.

```
Euphoria
    Deployment
      ✓ should initialize accounts with correct amounts
    Staking (big)
      ✓ should initialize with right initialSupply amount
(big)
      ✓ should createStake with the right amount (big)
      ✓ should unStake successfully with right amount after
one day (big)


  2 passing (1s)
  2 failing
```

Weaknesses

This section contains the list of discovered weaknesses.

1. Usage of ABDKMathQuad Can Lead to Loss of Rewards

Severity: Medium

Resolution: do not use the library

Description:

ABDKMathQuad library usage in the file **Contracts/blob/main/contracts/Staking.sol** can lead to lower rewards paid, since **ABDKMathQuad** implements the **IEEE 754** standard. The fraction is saved in the lower 112 bits, so an overflow can take place if the reward amount will be higher than 2^113 in full precision.

```
for(uint i = 0; i < userStakesCount; i++) {
        uint256 deposited = _stake[_stakeHolder][i];
        rewardInBytes = ABDKMathQuad.add(rewardInBytes,
ABDKMathQuad.mul(ABDKMathQuad.fromUInt(deposited),
ABDKMathQuad.sub(_distributedRewards,
_distributedRewardsSnapshot[_stakeHolder][i])));
        totalDeposited += deposited;
    }
```

2. Function Definition Does Not Correspond to Its Logic

Severity: Low

Resolution: change the function definition

Description:

The **getUserStakesCount()** function definition on line 137 in the file **Contracts/blob/main/contracts/Staking.sol** does not correspond to its logic. The definition should be changed to **getUserStakesAmount()**.

```
function getUserStakesCount(address stakeHolder) external view
returns(uint256)
```

3. Unnecessary Statement Used

Severity: Low

Resolution: remove the unnecessary statement

Description:

The **if()** statement on line 188 in the file **Contracts/blob/main/contracts/Staking.sol** seems to be unnecessary, since it always enters the block (except when **_distributionEnded** = true, and it stops the distribution altogether).

```
if(_previousTotalStakes == 0 || getCurrentDay() == _lastActiveDay) {
```

4. Token Balance Inconsideration

Severity: Low

Resolution: take token balances into consideration

Description:

The emergency function **finalize()** on line 206 in the file **Contracts/blob/main/contracts/Staking.sol** does not consider token balances and will only send the blockchain's native asset.

```
function finalize() external onlyOwner contractStarted
contractExpired {
    selfdestruct(payable(_msgSender()));
}
```

5. Unnecessary Function Call

Severity: Low

Resolution: remove the unnecessary call

Description:

The **_distributeRewards()** function on line 225 in the file **Contracts/blob/main/contracts/Staking.sol** is called twice.

```
if(!_distributionEnded)
        {
            if(getCurrentDay() != _lastActiveDay) {
                _distributeRewards();
            }

            _distributeRewards();
        }
```

6. Unnecessary Statement Used

Severity: Low

Resolution: remove the unnecessary check

Description:

The if() statement on line 284 in the file Contracts/blob/main/contracts/Staking.sol seems to be unnecessary, since the check is done on line 282 with a require() function.

```
require(_isStakeHolder == true, "Staking: There is not any stake
holder with provided address");

        if(_isStakeHolder) {
```

7. Missing Percentage Upper Boundary Check

Severity: Low

Resolution: add a check for the percentage to be <=100

Description:

A check should be added on line 298 in the file **Contracts/blob/main/contracts/Staking.sol** for the percentage to be <=100, since an admin can accidentally provide a percentage that is higher than 100.

```
require(supplyPercentage > 0, "Staking: Supply percentage cannot be a zero value");
```

8. Clean Code Recommendation

Severity: Low

Resolution: use 1 days alias

Description:

It is recommended to use **1 days** alias instead of **86400** seconds on lines 69 and 315 in the file **Contracts/blob/main/contracts/Staking.sol**. This will increase the readability of the code.

```
_startDate = block.timestamp - (block.timestamp % 86400);
return (block.timestamp - _startDate) / 86400;
```

9. Variable Visibility Not Set

Severity: Low

Resolution: state the variable visibility

Description:

The visibility of the **_unlockedTokensAmount, _maxTotalSupply and _isPaused** variables on lines 16, 17 and 24 respectively in the file **Contracts/blob/main/contracts/BridgeBsc.sol** is not set. Since the visibility is set in the case of other variables, it is recommended to set it for the above mentioned variables as well.

```
uint256 _unlockedTokensAmount
uint256 _maxTotalSupply
bool _isPaused
```

10. Misprint in Message

Severity: Low

Resolution: fix the misprint

Description:

The function **require()** on line 83 in the file **Contracts/blob/main/contracts/BridgeBsc.sol** contains a message with an incorrectly spelled word **“Unkown”**.

```
"BSC bridge: Unkown validator off-chain"
```

11. Variable Visibility Not Set

Severity: Low

Resolution: state the variable visibility

Description:

The visibility of the **_mintedTokensAmount, _maxTotalSupply and _contractStarted** variables on lines 23, 24 and 26 respectively in the file **Contracts/blob/main/contracts/BridgeEth.sol** is not set. Since the visibility is set in the case of other variables, it is recommended to set it for the above mentioned variables as well.

```
uint256 _mintedTokensAmount
uint256 _maxTotalSupply
bool _contractStarted
```

12. Misprint in Message

Severity: Low

Resolution: fix the misprint

Description:

The function **require()** on line 91 in the file **Contracts/blob/main/contracts/BridgeEth.sol** contains a message with an incorrectly spelled word **“Unkown”**.

```
"BSC bridge: Unkown validator off-chain"
```


hexens