



# expeee

A Secure Place For Web3

## **SMART CONTRACT AUDIT OF**

## **SmartAth Fair Launch**



**Contract Address** 

0x22b0E452551787dD5455a99AD417A3A983b1573b

| Page 1 | www.expelee.com





## **Audit Summary**

Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

**Audit Result: PASSED** 

**Ownership: NOT RENOUNCED** 

KYC Verification: Not done till date of audit

Audit Date: 15/07/2022

**Audit Team: EXPELEE** 

Be aware that smart contracts deployed on the blockchain aren't resistant to internal exploit, external vulnerability, or hack. For a detailed understanding of risk severity, source code vulnerability, functional hack, and audit disclaimer, kindly refer to the audit.

www.expelee.com | Page 2 |





## **DISCLAMER**

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always Do your own research and protect yourselves from being scammed. The Expelee team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools.

Under no circumstances did Expelee receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams.

This document should not be presented as a reason to buy or not buy any particular token. The Expelee team disclaims any liability for the resulting losses.

www.expelee.com Page 3 |



## **Contract Review**

Contract Name	SmartAth
Compiler Version	v0.8.15+commit.e14f2714
Optimization	Yes with 200 runs
License	Unlicense
Explorer	https://bscscan.com/address/0x22b0E 452551787dD5455a99AD417A3A983b1 573b#code
Symbol	SmartAth
Decimals	18
Total Supply	100,000,000
Domain	https://smartath.org/

www.expelee.com Page 4 |





# **Project Review**

Token Name: SmartAth

Web Site: https://smartath.org/

**Twitter: NOT FOUND** 

**Telegram: NOT FOUND** 

**Contract Address:** 

0x22b0E452551787dD5455a99AD417A3A983b1573b

**Platform: Binance Smart Chain** 

Token Type: BEP 20

Language: SOLIDITY

www.expelee.com | Page 5 |





## **Audit Methodology**

The scope of this report is to audit the smart contract source code. We have scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

## Category

- Unhandled Exceptions

- Transaction Order Dependency

Smart Contract Vulnerabilities - Integer Overflow

- Unrestricted Action

- Incorrect Inheritance Order

- Typographical Errors

- Requirement Violation

Source Code Review

- Gas Limit and Loops

- Deployment Consistency

- Repository Consistency

- Data Consistency

- Token Supply Manipulation

Functional Assessment - Operations Trail & Event Generation

- Assets Manipulation

- Liquidity Access

www.expelee.com | Page 6 |





# **Vulnerability Checklist**

Νō	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Re-entrancy. Cross-function raceconditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

www.expelee.com | Page 7 |

## **Manual Audit**

- Low-Risk
- 3 low-risk code issues found
  - Medium-Risk
- 0 medium-risk code issues found
  - High-Risk
  - 0 high-risk code issues found

www.expelee.com | Page 8 |



# **Audit Summary**

Compiled with solc

Number of lines: 1397 (+ 0 in dependencies, + 0 in tests)

Number of assembly lines: 0

Number of contracts: 12 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 15 Number of informational issues: 39

Number of low issues: 3 Number of medium issues: 0 Number of high issues: 0

ERCs: ERC20

+	+	<b></b>	+	+	+
Name	# functions	ERCS	ERC20 info	Complex code	Features
IUniswapV2Router02	24			No	Receive ETH
IUniswapV2Factory	8			No	
SignedSafeMath	4			No	
SafeMath	13			No	
SafeCast	14			No	
SmartAth	46	ERC20	No Minting	Yes	Receive ETH
			Approve Race Cond.		Send ETH
<del>+</del>	<b></b>		<u> </u>	<del></del>	<u> </u>

www.expelee.com | Page 9 |





#### 1) Contract contains Reentrancy vulnuerabilities

```
function _transfer(
        address from,
        address to,
        uint256 amount
    ) internal override {
        require(from != address(0), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");
        if(amount == 0) {
            super._transfer(from, to, 0);
            return;
        }
        uint256 contractTokenBalance = balanceOf(address(this));
        bool overMinTokenBalance = contractTokenBalance >= swapTokensAtAmount;
        if(
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            to == uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            contractTokenBalance = swapTokensAtAmount;
            swapAndLiquify(contractTokenBalance);
        }
        if(!_isExcludedFromFees[from] && !_isExcludedFromFees[to] ) {
            uint256 fees;
            if(from == uniswapV2Pair) {
              fees = amount.mul(buyFee).div(100);
```

#### Recommendation

Apply the check-effects-interaction pattern

www.expelee.com Page 10 |





#### 2) Local variable shadowing

Detection of shadowing using local variables.

```
constructor(address _owner) ERC20("SmartAth", "SmartAth") {
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
```

#### Recommendation

Rename the local variables that shadow another component .

www.expelee.com | Page 11 |



#### 3) Unused return

The return value of an external call is not stored in a local or state variable.

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    // add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        owner(),
        block.timestamp
    );
}
```

#### Recommendation

Ensure that all the return values of the function calls are used.

www.expelee.com | Page 12 |





## Manual Audit (Contract Function)

```
contract SmartAth is ERC20, Ownable {
   using SafeMath for uint256;
   IUniswapV2Router02 public uniswapV2Router;
    address public immutable uniswapV2Pair;
   bool private inSwapAndLiquify;
   bool public swapAndLiquifyEnabled = true;
    uint256 public swapTokensAtAmount = 50000 * (10**18);
   uint256 public buyFee = 5;
   uint256 public sellFee = 10;
   // exlcude from fees and max transaction amount
   mapping (address => bool) private _isExcludedFromFees;
    event ExcludeFromFees(address indexed account, bool isExcluded);
    event SwapAndLiquifyEnabledUpdated(bool enabled);
    event SwapAndLiquify(
        uint256 tokensIntoLiqudity,
        uint256 ethReceived
    );
   modifier lockTheSwap {
        inSwapAndLiquify = true;
        inSwapAndLiquify = false;
   }
    constructor(address owner) ERC20("SmartAth", "SmartAth") {
        IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
         // Create a uniswap pair for this new token
        address uniswapV2Pair = IUniswapV2Factory( uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());
        uniswapV2Router = uniswapV2Router;
        uniswapV2Pair = uniswapV2Pair;
        // exclude from paying fees
        excludeFromFees( owner, true);
        excludeFromFees(address(this), true);
```

www.expelee.com Page 16 |





## Important Points To Consider

- ✓ Verified contract source
- ✓ Token is sellable (not a honeypot) at this time
- X Ownership renounced or source does not contain an owner contract
  - X Source does not contain a fee modifier
    - ✓ Buy fee is less than 10% (5%)
    - ✓ Sell fee is less than 10% (9.9%)
- X Owner/creator wallet contains less than 10% of circulating token supply (29.93%)

www.expelee.com Page 17 |





## About Expelee

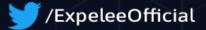
Expelee is a community driven organisation dedicated to fostering an antirug movement. We're here to keep investment safe from fraudsters. We've encountered several rug pulls and know how it feels to be duped, which is why we don't want anybody else to go through the same experience. We are here to raise awareness through our services so that the future of cryptocurrency can be rug-free.

The auditing process focuses to the following considerations with collaboration of an expert team:

- Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
- Manually detailed examination of the code line by line by experts.
- Live test by multiple clients using Test net.
- Analysing failure preparations to check how the Smart
- Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

### Social Media







www.expelee.com | Page 18 |