expelee

Building the Futuristic Blockchain Ecosystem

Security Audit Report

FOR

ATLAS

OVERVIEW

The 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	Not Passed
KYC Verification	Not Done
Audit Date	24 March 2023

NOTE

In our audit report, we have identified a critical centralization issue within the smart contract's farm function. This function currently checks whether the _totalSupply is less than the maximum supply, but fails to prevent the minting of arbitrary amounts. As a consequence, this oversight may expose the contract to potential exploits, allowing bad actors to mint tokens at their discretion. We highly recommend addressing this issue in order to maintain the integrity and security of the tokenomics system. This warning is of critical severity and should be prioritized in order to mitigate potential risks associated with the centralization of the minting process.

PROJECT DESCRIPTION

ATLAS

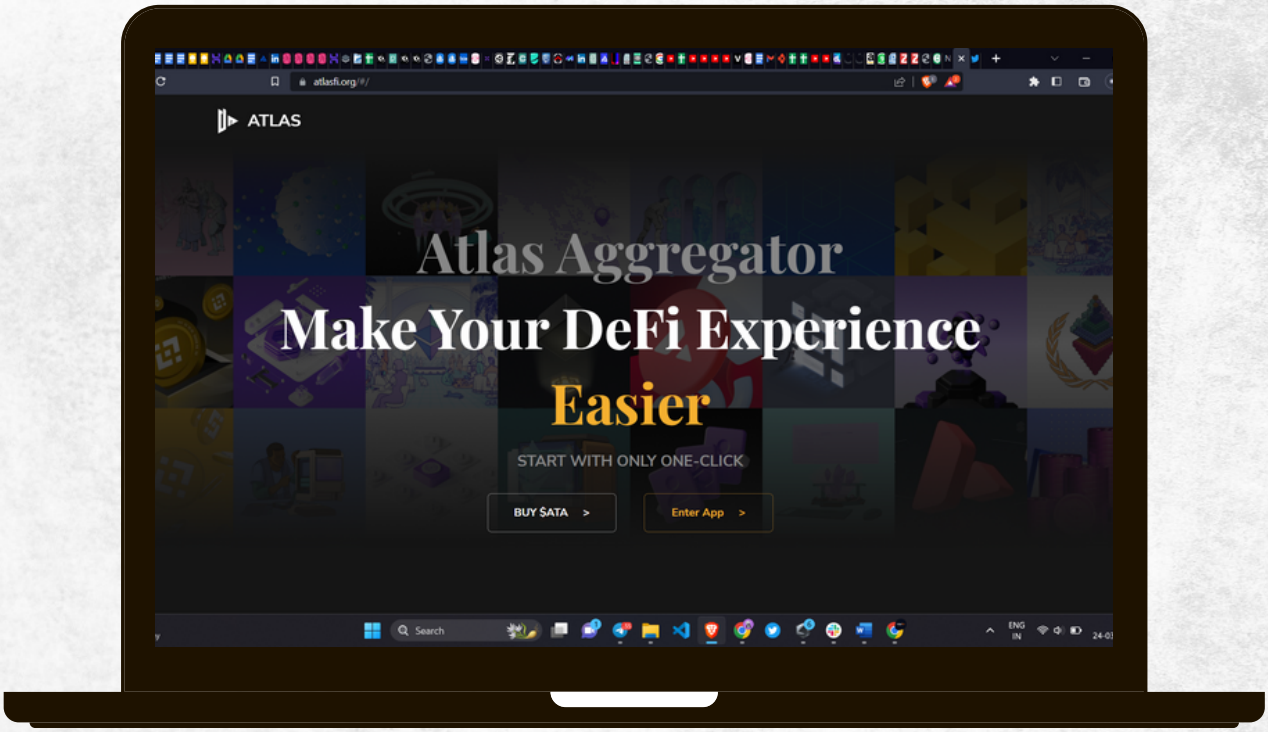
Atlas aggregator has been designed with security and community governance, and platform governance will be gradually passed on to the community.

expelee

expelee.com

Social Media Profiles

ATLAS



https://atlasfi.org/

https://t.me/Atlasaggregator

https://twitter.com/atlasaggregator

It's always good to check the social profiles of the project, before making your investment.

-Team Expelee

CONTRACT DETAILS

Token Name
ATLAS

Symbol
ATA

Network
Arbitrum

Language
Solidity

Contract Address (Verified)
0x296A0b8847BD4ED9af71a9ef238fa5Be0778B611

Token Type
ERC20

Total Supply
15,000,000

Contract SHA-256 Checksum:
cf119fbc742a4369bb1e9bb033a98273d7b68b00

Owner Wallet
0x460683B6bDC8a807d116273d9AcdFb3d6cfE88b5

Deployer Wallet
0x460683B6bDC8a807d116273d9AcdFb3d6cfE88b5

AUDIT METHODOLOGY

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

Tools

- DE
- Open Zeppelin
- Code Analyzer
- Solidity Code
- Complier
- Hardhat

FUNCTION OVERVIEW

Can Take Back Ownership	Not Detected
Owner Change Balance	Not Detected
Blacklist	Not Detected
Modify Fees	Not Detected
Proxy	Not Detected
Whitelisted	Not Detected
Anti Whale	Not Detected
Trading Cooldown	Not Detected
Transfer Pausable	Not Detected
Cannot Sell All	Not Detected
Hidden Owner	Not Detected
Mint	Detected

VULNERABILITY CHECKLIST

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions & reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

AUDIT SUMMARY

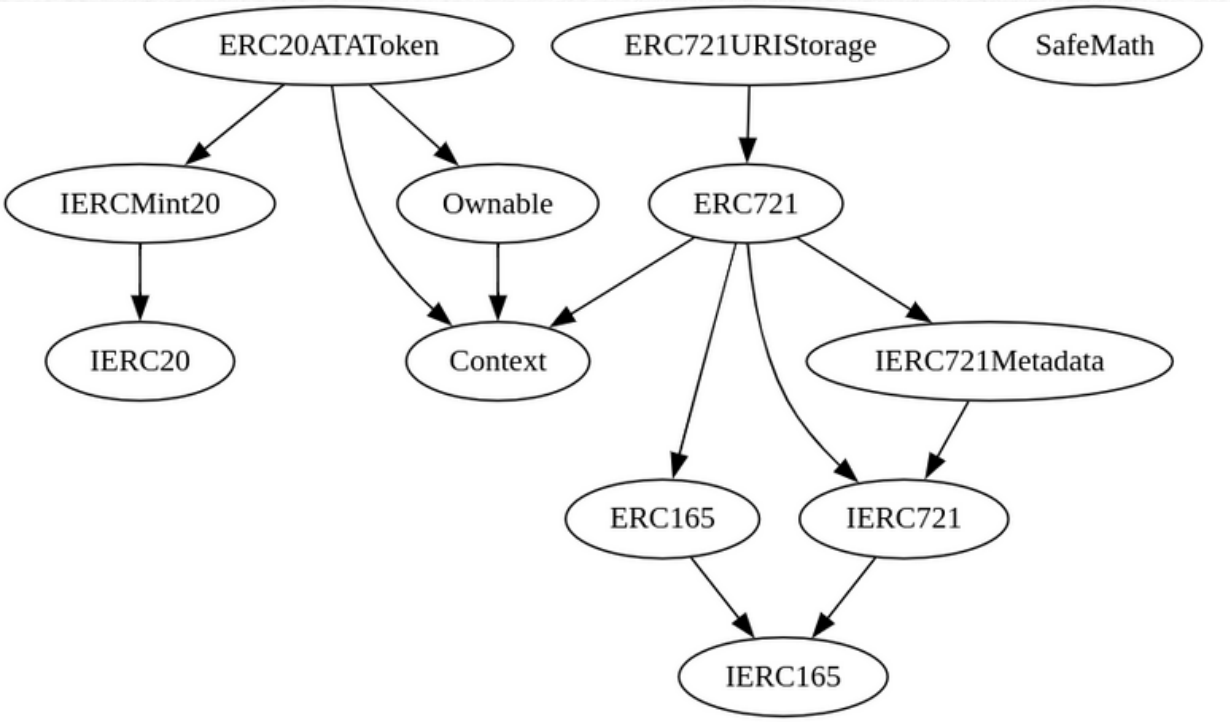
Used Tools:

1.Manual Review: The code has undergone a line-by-line review by the Expelee team.

2.1.BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3.Slither: The code has undergone static analysis using Slither.

Inheritance Trees:



Summary:

- **Owner is able to mint new tokens**
- Owner is not able to set buy/sell/transfer fees
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

Functional Tests

0xD99D1c33F9fC3444f8101754aBC46c52416550D1
All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xd84d7d3661c59d4bbcf2346fe2ea5eff8ea56f3802bd8ad8179fd49f3a7c1778

2- Buying (0% tax) (passed):
https://testnet.bscscan.com/tx/0x60913254b80dc188b7b0334d7ab78980e2727a27a83ba7e3234e32e6b58a6ce6

3- Selling (0% tax) (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2ce914f1a7c03db806c576b3828dc7e14026a1c1def5c19d83b701d48854b9de

4- Transferring (0% tax) (0% tax) (passed):
https://testnet.bscscan.com/tx/0x1a888bfec893fa4807ee51c3ae8f2f6558aca64e980e3a1f5179bd8f8cf0d5f3

Functional Tests

5- Farm (Failed):
farm failed because undesired amounts can be minted, in this transaction
115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,549,039,457
new tokens were minted:
https://testnet.bscscan.com/tx/0x7b610cbc4f9286cb384343c165a68c0fe4db33773e2c9a11acdb77962b420e2d

6- Burning (passed):
https://testnet.bscscan.com/tx/0x5f245aa38d8d8f58864f43cb804205712e731eb9f658bf9380364cce8a2eeceb

MANUAL AUDIT

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to a methodology based on OWASP standards.

Vulnerabilities are divided into three primary risk categories: high, medium, and low.

High-level considerations for vulnerabilities span the following key areas when conducting assessments:

- Malicious Input Handling
- Escalation of privileges
- Arithmetic
- Gas use

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

FINDINGS

- Critical Risk Findings:2
- High Risk Findings:0
- Medium Risk Findings:0
- Low Risk Findings:0
- Suggestions & discussion: 1
- Gas Optimizations : 0

Critical Risk Findings

Centralization - Wrong condition allows unlimited mint

Severity: Critical
Function: farm
Lines: 382
Overview:

The present implementation of the farm function verifies if the _totalSupply is below the maximum supply. However, this does not prevent the minting of arbitrary amounts.

```
function farm(address to, uint256 amount) public override returns (bool) {
    require(msg.sender == farmAddress);
    if (_totalSupply > 100000000 * 1e18) {
        return false;
    }
    _mint(to, amount);
    return true;
}
```

Recommendation:
change the condition to below implmentation, prevent undesired mint amounts:

```
function farm(address to, uint256 amount) public override returns (bool) {
    require(msg.sender == farmAddress);
    if (_totalSupply + amount > 100000000 * 1e18) {
        return false;
    }
    _mint(to, amount);
    return true;
}
```

Critical Risk Findings

Centralization - Minting new tokens:

Severity: Critical
Function: farm
Lines: 382
Overview:

The token contract includes a minting function that allows the creation of new tokens. While this function is necessary for rewarding stakers, the current implementation possesses a critical vulnerability that could potentially be exploited.
farm function can be used by **farmAddress** to mint up to 6x initial supply **(100000000 max supply).**
This function is suppose to be only called by farming contract, however, owner of ATA token is able to change **farmAddress** to any other wallet, this means the new wallet will be able to mint new tokens

```
function farm(address to, uint256 amount) public override returns (bool) {
    require(msg.sender == farmAddress);
    if (totalSupply > 100000000 * 1e18) {
        return false;
    }
    _mint(to, amount);
    return true;
}
```

```
function setFarmAddress(address farmAddress_) public onlyOwner {
    farmAddress = farmAddress_;
}
```

Critical Risk Findings

Recommendation:

•Renouncing ownership will mitigate this issue, removing the ability for the token owner to arbitrarily change the farmAddress.
•Implement a governance model for upgrading the farming contract, ensuring that decisions are made by a majority of stakeholders.
•Delete the setFarmAddress function, removing the ability to change the farmAddress once it has been initially set.

Suggestions & Discussion

Gas Optimization – Constant & Immutable variables:

Severity: Informational
Type: Logical
Overview:

Some variables can be declared as immutable or constant in order to reduce gas usage when this variables are accesses later.

Recommendation:
Declare this vairables as constant:
_decimals
_name
_symbol

ABOUT EXPELEE

Expelee is a product-based aspirational Web3 Start-up. Coping up with numerous solutions for blockchain Security and constructing a Web3 Ecosystem from Deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.

www.expelee.com

expeleeofficial

expelee

Expelee

expelee

expelee_official

expelee-co

expelee

Building the Futuristic **Blockchain Ecosystem**

DISCLAIMER

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.