

# expelee

Building the Futuristic **Blockchain Ecosystem**

# Audit Report FOR



## Aark Protocol

# 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	Passed
 KYC Verification	Not Done
 Audit Date	10 Oct 2022

# PROJECT DESCRIPTION

## Aark Protocol

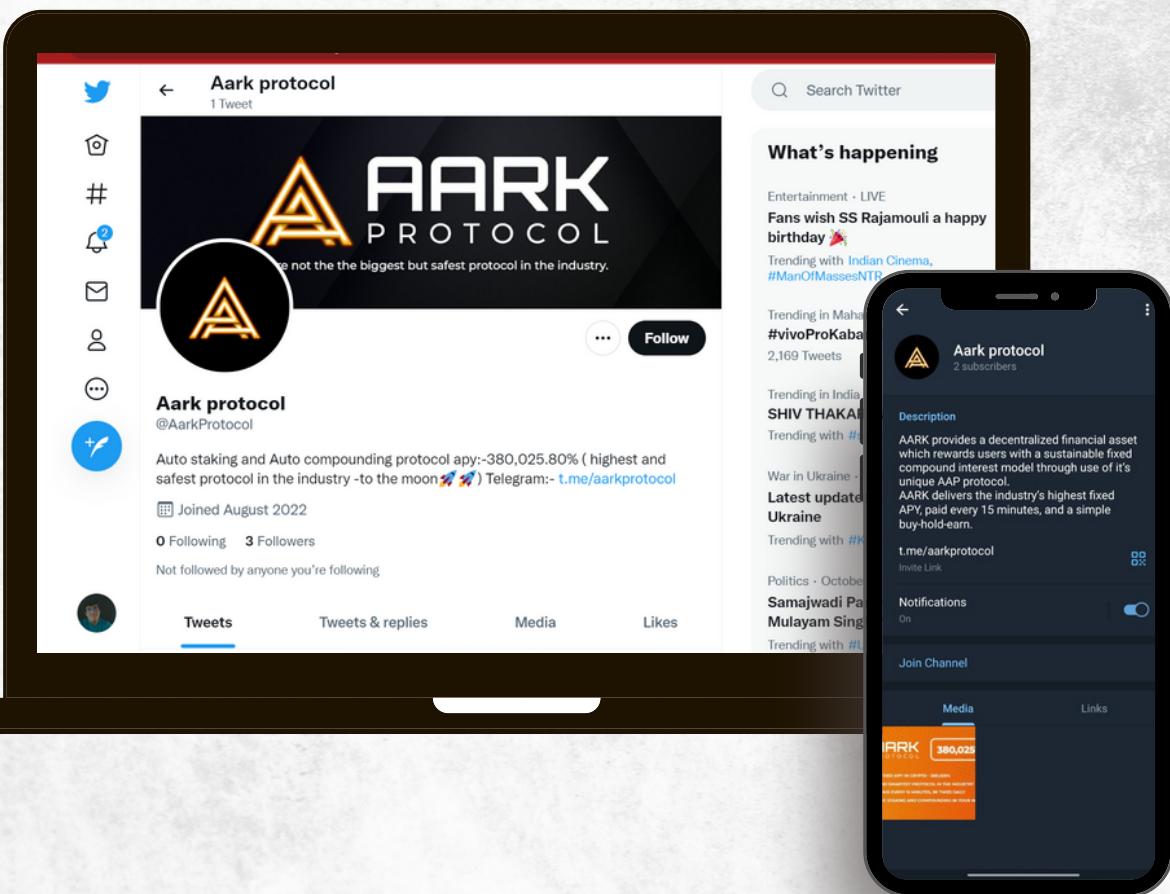
AARK provides a decentralized financial asset which rewards users with a sustainable fixed compound interest model through use of it's unique AAP protocol.

AARK delivers the industry's highest fixed APY, paid every 15 minutes, and a simple buy-hold-earn.



# Social Media Profiles

## Aark Protocol



No Website

<https://t.me/aarkprotocol>

<https://t.me/aarkprotocol>

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

-Team Expelee

# CONTRACT DETAILS

Token Name

**AARK**

---

Symbol

**Arkp**

---

Network

**BSC**

---

Language

**Solidity**

---

Contract Address (Verified)

**0x998B8b4E14a0437d65007b6A0fA2941aB520aF1F**

---

Token Type

**ERC 20**

---

Total Supply

**5,500,000**

---

Compiler

**v0.7.6+commit.7338295f**

---

License

**default**

---

Contract SHA-256 Checksum:

**83ffc04f247960807720f9edcc8773247ecaf8c06f46615c7e6bb3e35ce071dd**

---

# 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	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	Not 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.

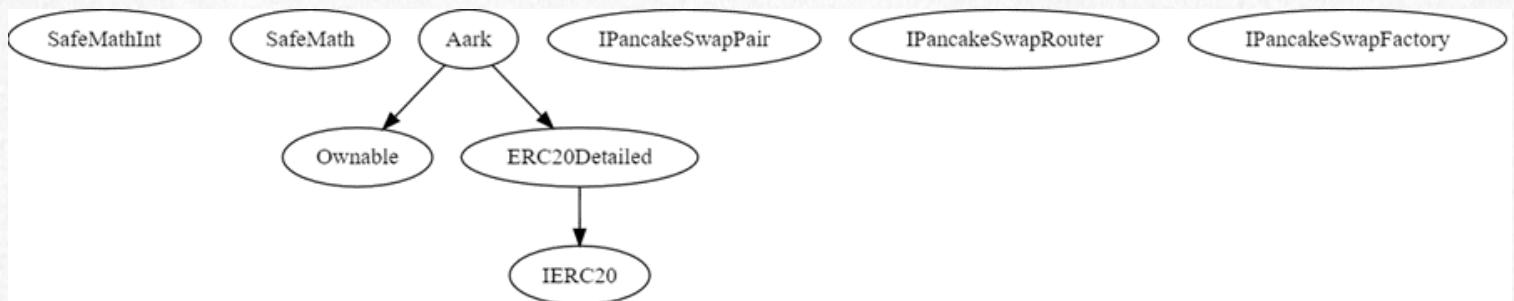
# AUDIT SUMMARY

## Ownership:

0x7dec37cc8681e4af23fb9fa8e7a69589e782f939, is the owner of the contract

## Contracts & Inheritance Tree:

all of below contracts are in this audit scope



## Summary

- Auto Staking Token, by holding AARK your balance will automatically increase (380,025% APY)
- Owner is able to blacklist an arbitrary address from buying/selling/transferring tokens
- there is a static 15% tax on buys/sells.
- owner is not able to disable trading

# 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 Summary

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

## High Risk Findings

**Centralization** - owner is able to blacklist an arbitrary wallet from buying/selling/transfer.

```
function setBotBlacklist(address _botAddress, bool _flag)
external onlyOwner {
    require(isContract(_botAddress), "only contract address,
not allowed externally owned account");
blacklist[_botAddress] = _flag;
}
```

**suggestion:** define action that might blacklist a wallet, or delete this functionality

## Medium Risk Findings

**Logical - use of outdated compiler : using outdated compiler can lead to underflow/overflow/unexpected bugs.**

**suggestion: use compiler > 0.8.0**

## Gas Optimizations

- **only import needed functions from pancakeswap interface to reduce contract size**
- **specify zero , dead, feeDenominator vairables as constant to save gas**

## Suggestions

- **use compiler > 0.8.0**

# 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](http://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.