auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

# Smart Contract Audit

FOR

# Audinals

DATED : 1 September 23'

# MANUAL TESTING

## Centralization – Blacklisting

**Severity: High**
**function: transferProtection**
**Status: Open**
**Overview:**
**Owner is able to disable sell and wallet to wallet transfers for an arbitrary wallet by using transferProtection function.**

```
function transferProtection(
address[] calldata _wallets,
uint256 _enabled
  ) external onlyOwner {
for (uint256 i = 0; i < _wallets.length; i++) {
walletProtection[_wallets[i]] = _enabled;
}
  }

function _beforeTokenTransfer(address from, address to) internal
view {
require(
walletProtection[from] == 0 || to == owner(),
"Wallet protection enabled, please contact support"
);
  }
```

**Blacklisting liquidity pool using this function can blacklist buy transactions.**

**Suggestion**
**Implement an automated method for blacklisting malicious wallets and define all the actions that might blacklist a wallet**

# MANUAL TESTING

## Centralization – Withdrawing dividend tracker rewards

**Severity: High**
**function: setDistirbutor**
**Status: Open**
**Overview:**
**Owner is able to withdraw accumulated reward tokens from dividend tracker using setDistributor function. If "migrate" Is set to true, all reward tokens will be sent to new distributor contract.**

```
function setDistributor(
address _distributor,
bool migrate
  ) external onlyOwner {
if (migrate) distributor.migrate(_distributor);

distributor = IDividendDistributor(_distributor);
distributor.initialize();
  }
```

**Suggestion**
**Created a timelock contract which delays updating of dividend tracker or make sure that rewards are not withdrawable**

# MANUAL TESTING

## Centralization – Updating dividend tracker

**Severity: High**
**function: setDistributor**
**Status: Open**
**Overview:**
**setDistributor function allows owner to update dividend tracker to a new contract. Updating dividend tracker to a malicious contract could result in unexpected behaviour while trying to transfer / buy / sell tokens.**

```
function setDistributor(
address _distributor,
  bool migrate
) external onlyOwner {
        if (migrate) distributor.migrate(_distributor);
        distributor = IDividendDistributor(_distributor);
        distributor.initialize();
}
```

**Suggestion**
**Since dividend tracker is one of core components of the contract, updating it to a malicious address could affect transfer/buy/sell transactions. Its suggested to implement a more decentralized method for updating dividend tracker such as using a governance model or a time lock contract.**

# AUDIT SUMMARY

**Project name** – Audinals

**Date**: 1 September 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status: FAILED**

## Issues Found

| Status | Critical | High | Medium | Low | Suggestion |
|---|---|---|---|---|---|
| Open | 0 | 3 | 0 | 0 | 0 |
| Acknowledged | 0 | 0 | 0 | 0 | 0 |
| Resolved | 0 | 0 | 0 | 0 | 0 |

# USED TOOLS

## Tools:

**1- Manual Review:**
A line by line code review has been performed by audit ace team.

**2- 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.

**Testnet version:**
The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
https://testnet.bscscan.com/address/0x9929f3cBE5c401Ab4be9c1504829B4E16a1E7de5

# Token Information

**Token Address :**
0x5F68F6e8Da909e3d1Ed3c1d28553563bADE5BB4a

**Name:** Audinals

**Symbol:** AUDO

**Decimals:** 9

**Network:** Ethereum

**Token Type:** ERC20

**Owner:** 0x389346E15bd2D4CFB046E1C70911Dc1D9b9B639B

**Deployer:**
0x389346E15bd2D4CFB046E1C70911Dc1D9b9B639B

**Token Supply:** 1,000,000,000

**Checksum:**
de8a0e13482feb8a1e391439f8e92323e706c58c

**Testnet version:**
The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
**https://testnet.bscscan.com/address/0x9929f3cBE5c401Ab4be9c1504829B4E16a1E7de5**

# TOKEN OVERVIEW

**buy fee: 5%**

**Sell fee: 5%**

**transfer fee: 0%**

**Fee Privilege: Owner**

**Ownership: Owned**

**Minting: None**

**Max Tx: No**

**Blacklist: yes**

**Other Privileges:**

**- Initial distribution of the tokens**

**- Updating dividend tracker**

**- limiting wallets from sell/transfers**

# AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

# VULNERABILITY CHECKLIST

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

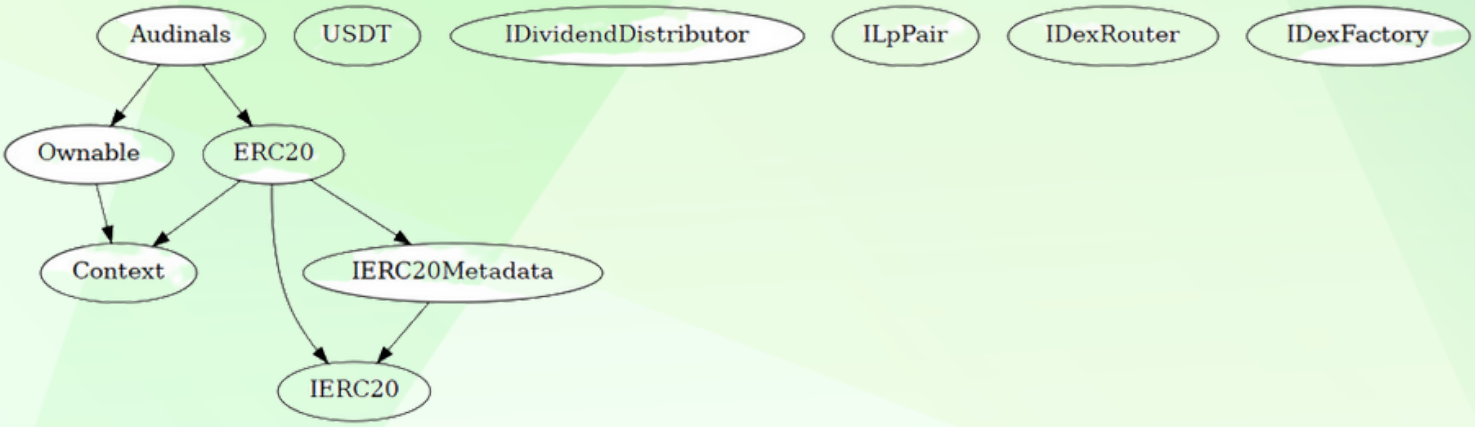
# CLASSIFICATION OF RISK

| Severity | Description |
| --- | --- |
| Critical | These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away. |
| High-Risk | A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way. |
| Medium-Risk | A vulnerability that could affect the desired outcome of executing the contract in a specific scenario. |
| Low-Risk | A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective. |
| Gas Optimization /Suggestion | A vulnerability that has an informational character but is not affecting any of the code. |

# Findings

| Severity | Found |
| --- | --- |
| Critical | 0 |
| High-Risk | 3 |
| Medium-Risk | 0 |
| Low-Risk | 0 |
| Gas Optimization / Suggestions | 0 |

# INHERITANCE TREE



# POINTS TO NOTE

**- Owner is not able to change current fees (5% buy/sell and 0% transfer)**

- Owner is able to blacklist an arbitrary wallet (blacklisted wallets are not capable of selling/transferring)

- Owner is not able to mint new tokens

- Owner is not able to set maximum wallet and maximum buy/sell/transfer limits

**- Owner must enable trades manually (trades already enabled)**

# STATIC ANALYSIS

```
Reentrancy in Audinals.airdropToWallets(address[],uint256[],bool) (contracts/Token.sol#771-794):
        External calls:
        - distributor.setShare(wallets[i],amountsInTokens[i] * _decimalFactor) (contracts/Token.sol#788-791)
        Event emitted after the call(s):
        - Transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - super._transfer(msg.sender,wallets[i],amountsInTokens[i] * _decimalFactor) (contracts/Token.sol#782-786)
Reentrancy in Audinals.prepare(uint256) (contracts/Token.sol#666-690):
        External calls:
        - lpPair = IDexFactory(dexRouter.factory()).createPair(ETH,address(this)) (contracts/Token.sol#673-676)
        Event emitted after the call(s):
        - Transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - super._transfer(msg.sender,address(this),tokens * _decimalFactor) (contracts/Token.sol#680)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Audinals._transfer(address,address,uint256) (contracts/Token.sol#555-614) has a high cyclomatic complexity (17).
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cyclomatic-complexity
INFO:Detectors:
Context._msgData() (contracts/Token.sol#21-24) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.17 (contracts/Token.sol#14) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Audinals.withdrawTax() (contracts/Token.sol#650-659):
        - (success,None) = address(msg.sender).call{value: address(this).balance}() (contracts/Token.sol#656-658)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IDexRouter.WETH() (contracts/Token.sol#372) is not in mixedCase
Parameter Audinals.updateSplit(uint256)._split (contracts/Token.sol#661) is not in mixedCase
Parameter Audinals.setDistributor(address,bool)._distributor (contracts/Token.sol#698) is not in mixedCase
Parameter Audinals.setTaxCollector(address)._collector (contracts/Token.sol#707) is not in mixedCase
Parameter Audinals.setDistributionCriteria(uint256,uint256,uint256)._minPeriod (contracts/Token.sol#712) is not in mixedCase
Parameter Audinals.setDistributionCriteria(uint256,uint256,uint256)._minDistribution (contracts/Token.sol#713) is not in mixedCase
Parameter Audinals.setDistributionCriteria(uint256,uint256,uint256)._claimAfter (contracts/Token.sol#714) is not in mixedCase
Parameter Audinals.transferProtection(address[],uint256)._wallets (contracts/Token.sol#797) is not in mixedCase
Parameter Audinals.transferProtection(address[],uint256)._enabled (contracts/Token.sol#798) is not in mixedCase
Constant Audinals._decimals (contracts/Token.sol#442) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Audinals._decimalFactor (contracts/Token.sol#443) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (contracts/Token.sol#22)" inContext (contracts/Token.sol#16-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Slither:./contracts/Token.sol analyzed (11 contracts with 88 detectors), 41 result(s) found
```

**Result => A static analysis of contract's source code has been performed using slither,**

**No major issues were found in the output**

# CONTRACT ASSESMENT

| Contract| Type |Bases | | |
|:---------:|:----------:|:---------:|:---------:|:---------:|
| └| **Function Name** |**Visibility** | **Mutability** |**Modifiers** |
||||||
| **Context** | Implementation | ||| |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
||||||
| **IERC20** | Interface | ||| |
| └ | totalSupply | External ❗ | |NO ❗ | |
| └ | balanceOf | External ❗ | |NO ❗ | |
| └ | transfer | External ❗ | 🛑 |NO ❗ | |
| └ | allowance | External ❗ | |NO ❗ | |
| └ | approve | External ❗ | 🛑 |NO ❗ | |
| └ | transferFrom | External ❗ | 🛑|NO ❗ | |
||||||
| **IERC20Metadata** | Interface | IERC20 ||| |
| └ | name | External ❗ | |NO ❗ | |
| └ | symbol | External ❗ | |NO ❗ | |
| └ | decimals | External ❗ | |NO ❗ | |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata ||| |
| └ | <Constructor> | Public ❗ | 🛑|NO ❗ | |
| └ | name | Public ❗ | |NO ❗ | |
| └ | symbol | Public ❗ | |NO ❗ | |
| └ | decimals | Public ❗ | |NO ❗ | |
| └ | totalSupply | Public ❗ | |NO ❗ | |

# CONTRACT ASSESMENT

| └ | balanceOf | Public ❗ | |NO❗ |
| └ | transfer | Public ❗ | 🛑 |NO❗ |
| └ | allowance | Public ❗ | |NO❗ |
| └ | approve | Public ❗ | 🛑 |NO❗ |
| └ | transferFrom | Public ❗ | 🛑|NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑|NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑|NO❗ |
| └ | _transfer | Internal 🔒 | 🛑| |
| └ | _approve | Internal 🔒 | 🛑| |
| └ | _initialTransfer | Internal 🔒 | 🛑 | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🛑|NO❗ |
| └ | owner | Public ❗ | |NO❗ |
| └ | renounceOwnership | Public ❗ | 🛑| onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑| onlyOwner |
||||||
| **USDT** | Interface | |||
| └ | balanceOf | External ❗ | 🛑 |NO❗ |
| └ | transfer | External ❗ | 🛑 |NO❗ |
| └ | approve | External ❗ | 🛑 |NO❗ |
||||||
| **IDividendDistributor** | Interface ||||
| └ | initialize | External ❗ | 🛑|NO❗ |
| └ | setDistributionCriteria | External ❗ | 🛑 |NO❗ |
| └ | setShare | External ❗ | 🛑 |NO❗ |
| └ | deposit | External ❗ | 💵 |NO❗ |
| └ | claimDividend | External ❗ | 🛑|NO❗ |
| └ | getUnpaidEarnings | External ❗ | |NO❗ |
| └ | getPaidDividends | External ❗ | |NO❗ |
| └ | getTotalPaid | External ❗ | |NO❗ |
| └ | getClaimTime | External ❗ | |NO❗ |
| └ | getLostRewards | External ❗ | |NO❗ |
| └ | getTotalDividends | External ❗ | |NO❗ |
| └ | getTotalDistributed | External ❗ ||NO❗ |
| └ | getTotalSacrificed | External ❗ | |NO❗ |
| └ | countShareholders | External ❗ | |NO❗ |
| └ | migrate | External ❗ | 🛑 |NO❗ |
||||||
| **ILpPair** | Interface | |||

# CONTRACT ASSESMENT

| └ | sync | External ❗ | 🛑 |NO❗ |
||||||
| **IDexRouter** | Interface | |||
| └ | factory | External ❗ | |NO❗ |
| └ | WETH | External ❗ | |NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO❗ |
| └ | swapExactETHForTokens | External ❗ | 💵 |NO❗ |
| └ | swapETHForExactTokens | External ❗ | 💵 |NO❗ |
| └ | addLiquidityETH | External ❗ | 💵 |NO❗ |
| └ | getAmountsOut | External ❗ | |NO❗ |
||||||
| **IDexFactory** | Interface | |||
| └ | createPair | External ❗ | 🛑|NO❗ |
||||||
| **Audinals** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑| ERC20 |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |
| └ | decimals | Public ❗ | |NO❗ |
| └ | updateSwapTokens | External ❗ | 🛑| onlyOwner |
| └ | toggleSwap | External ❗ | 🛑 | onlyOwner |
| └ | setPair | External ❗ | 🛑 | onlyOwner |
| └ | getSellFees | Public ❗ | |NO❗ |
| └ | getBuyFees | Public ❗ | |NO❗ |
| └ | excludeFromFees | Public ❗ | 🛑| onlyOwner |
| └ | setDividendExempt | External ❗ | 🛑| onlyOwner |
| └ | _transfer | Internal 🔒 | 🛑| |
| └ | swapTokensForEth | Private 🔐 | 🛑 | |
| └ | swapBack | Private 🔐 | 🛑| |
| └ | withdrawTax | External ❗ | 🛑|NO❗ |
| └ | updateSplit | External ❗ | 🛑 | onlyOwner |
| └ | prepare | External ❗ | 💵 | onlyOwner |
| └ | launch | External ❗ | 🛑 | onlyOwner |
| └ | setDistributor | External ❗ | 🛑| onlyOwner |
| └ | setTaxCollector | External ❗ | 🛑| onlyOwner |

# CONTRACT ASSESMENT

| └ | setDistributionCriteria | External ❗ | 🛑 | onlyOwner |
| └ | manualDeposit | External ❗ | 💵 |NO ❗ |
| └ | getPoolStatistics | External ❗ | |NO ❗ |
| └ | myStatistics | External ❗ | |NO ❗ |
| └ | checkClaimTime | External ❗ | |NO ❗ |
| └ | claim | External ❗ | 🛑 |NO ❗ |
| └ | airdropToWallets | External ❗ | 🛑 | onlyOwner |
| └ | transferProtection | External ❗ | 🛑| onlyOwner |
| └ | _beforeTokenTransfer | Internal 🔒 || |

### Legend

| Symbol| Meaning |
|:--------:|-----------|
| 🛑| Function can modify state |
| 💵| Function is payable |

# FUNCTIONAL TESTING

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xdb12f01790dff3c7cf322714e05ae34d976858a2d1662e1c22976c14fee46469

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x8fa4c31ca0a1c05fec856a80580e5b52da0c15d45a02a8b6421209a1b6cfac0b

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x72a29b8569ae2d3ecaa5cde56cca6d3367d6f6ee7cba90e74f64ccc72d4adc98

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x58dc91c835bff5dcf9ee1f8672333a33538d4b89abed207d148dda9acd0d677a

5- Buying when not excluded from fees (tax 5%) (passed):
https://testnet.bscscan.com/tx/0xe2611023093d4d84e5c61ace959ed305e551ea48674b8762f9ea296a985e82a4

6- Selling when not excluded from fees (tax 5%) (passed):
https://testnet.bscscan.com/tx/0x3a9bb20a0f73121dea26605c50fc251d0f091788bbd2c20cbd95d1d765c37d46

7- Transferring when not excluded from fees (0% tax ) (passed):
https://testnet.bscscan.com/tx/0x8e410fa858665ca35f1fdf30ef4c4e87b0ca9a819fa5c61b4155a52930fcc16d

8- Internal swap (ETH sent to dividend tracker) (passed):
https://testnet.bscscan.com/tx/0x3a9bb20a0f73121dea26605c50fc251d0f091788bbd2c20cbd95d1d765c37d46

# MANUAL TESTING

## Centralization – Blacklisting

**Severity: High**
**function: transferProtection**
**Status: Open**
**Overview:**
**Owner is able to disable sell and wallet to wallet transfers for an arbitrary wallet by using transferProtection function.**

```
function transferProtection(
address[] calldata _wallets,
uint256 _enabled
  ) external onlyOwner {
for (uint256 i = 0; i < _wallets.length; i++) {
walletProtection[_wallets[i]] = _enabled;
}
  }

function _beforeTokenTransfer(address from, address to) internal
view {
require(
walletProtection[from] == 0 || to == owner(),
"Wallet protection enabled, please contact support"
);
  }
```

**Blacklisting liquidity pool using this function can blacklist buy transactions.**

**Suggestion**
**Implement an automated method for blacklisting malicious wallets and define all the actions that might blacklist a wallet**

# MANUAL TESTING

## Centralization – Withdrawing dividend tracker rewards

**Severity: High**
**function: setDistirbutor**
**Status: Open**
**Overview:**
**Owner is able to withdraw accumulated reward tokens from dividend tracker using setDistributor function. If "migrate" Is set to true, all reward tokens will be sent to new distributor contract.**

```
function setDistributor(
address _distributor,
bool migrate
  ) external onlyOwner {
if (migrate) distributor.migrate(_distributor);

distributor = IDividendDistributor(_distributor);
distributor.initialize();
  }
```

**Suggestion**
**Created a timelock contract which delays updating of dividend tracker or make sure that rewards are not withdrawable**

# MANUAL TESTING

## Centralization – Updating dividend tracker

**Severity: High**
**function: setDistributor**
**Status: Open**
**Overview:**
**setDistributor function allows owner to update dividend tracker to a new contract. Updating dividend tracker to a malicious contract could result in unexpected behaviour while trying to transfer / buy / sell tokens.**

```
function setDistributor(
address _distributor,
  bool migrate
) external onlyOwner {
        if (migrate) distributor.migrate(_distributor);
        distributor = IDividendDistributor(_distributor);
        distributor.initialize();
}
```

**Suggestion**
**Since dividend tracker is one of core components of the contract, updating it to a malicious address could affect transfer/buy/sell transactions. Its suggested to implement a more decentralized method for updating dividend tracker such as using a governance model or a time lock contract.**

# 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 Auditace 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 Auditace 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 Auditace team disclaims any liability for the resulting losses.

# ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.


**https://auditace.tech/**

**https://t.me/Audit_Ace**

**https://twitter.com/auditace_**

**https://github.com/Audit-Ace**