



Smart Contract Audit

FOR
\$STOPG

DATED : 22 MAY 23'



AUDIT SUMMARY

Project name – \$TOPG

Date: 22 May, 2023

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

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

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

2. ETH Test Network: All tests were conducted on the ETH 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/token/0xf8Bd1cf071547584824B41ff57255CC3B70796B9>



Token Information

Name : TopG CEO

Symbol : \$TOPG

Decimals: 9

Network: BSC

Token Type: BEP20

Token Address:

0x929Dbd09EB827C9Ce2bFe1574783d61f7D3bB37
5

Owner:

0xb345c34aa77cBFc255B544ECCe12E1CD675A5eF
6 (at time of writing the audit)

Deployer: 0xb345c34aa77cBFc255B544ECCe12E1CD
675A5eF6



Token Information

Fees:

Buy Fees: 8%

Sell Fees: 8%

Transfer Fees: 0%

Fees Privilege: Constant fees

Ownership :

0xb345c34aa77cBFc255B544ECCe12E1CD675A5eF6

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges:- - changing swap threshold

- excluding wallets from fees

- including wallets in fees

- initial distribution of the tokens



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 is exercised 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

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

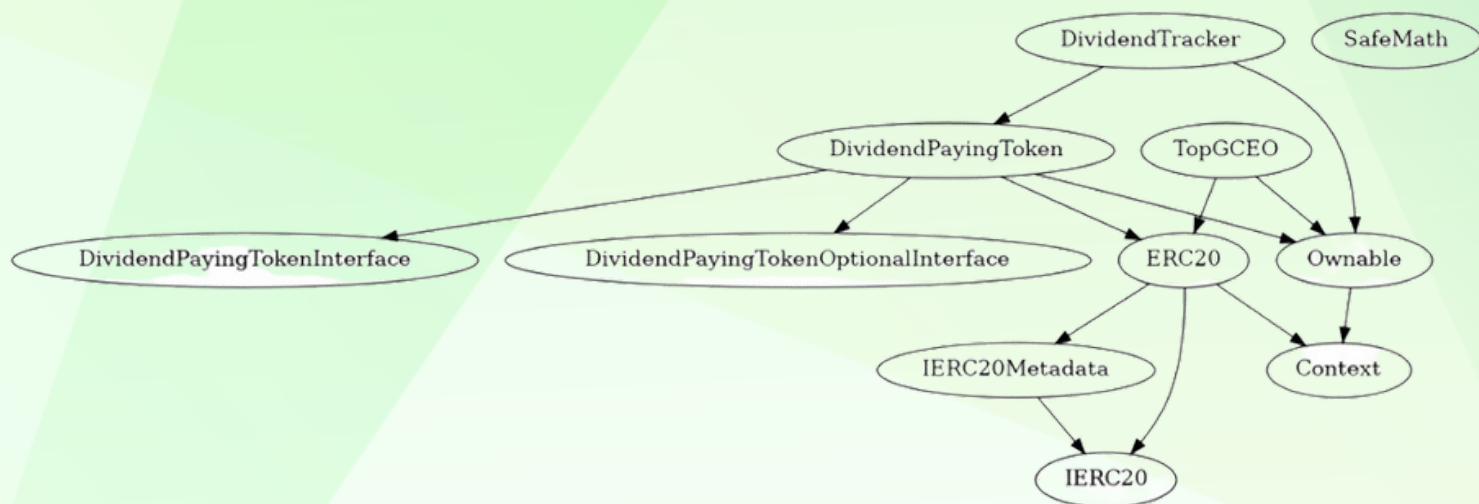
Severity

Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	1



INHERITANCE TREE





POINTS TO NOTE

- Owner is not able to change buy/sell/transfer fee (8% for buy and sell, 0% for transfers)
 - Owner is not able to blacklist an arbitrary address.
 - Owner is not able to disable trades
 - Owner is not able to set limits for buy/sell/transfer amounts
 - Owner is not able to mint new tokens
-



CONTRACT ASSESMENT

Contract	Type	Bases			
└──	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
Context Implementation					
└─	<Constructor>	Public !	●	NO !	
└─	_msgSender	Internal 🔒			
└─	_msgData	Internal 🔒			
Ownable Implementation Context					
└─	<Constructor>	Public !	●	NO !	
└─	owner	Public !		NO !	
└─	renounceOwnership	Public !	●	onlyOwner	
└─	transferOwnership	Public !	●	onlyOwner	
└─	_setOwner	Private 🔒	●		
SafeMath Library					
└─	add	Internal 🔒			
└─	sub	Internal 🔒			
└─	sub	Internal 🔒			
└─	mul	Internal 🔒			
└─	div	Internal 🔒			
└─	div	Internal 🔒			
└─	mod	Internal 🔒			
└─	mod	Internal 🔒			
SafeMathInt Library					
└─	mul	Internal 🔒			
└─	div	Internal 🔒			
└─	sub	Internal 🔒			
└─	add	Internal 🔒			
└─	abs	Internal 🔒			
└─	toUint256Safe	Internal 🔒			
SafeMathUint Library					
└─	toInt256Safe	Internal 🔒			
IterableMapping Library					
└─	get	Internal 🔒			
└─	getIndexOfKey	Internal 🔒			
└─	getKeyAtIndex	Internal 🔒			
└─	size	Internal 🔒			
└─	set	Internal 🔒	●		



CONTRACT ASSESMENT

```
| L | remove | Internal | 🔒 | ● | |
|||||
| **IFactoryV2** | Interface | |||
| L | getPair | External | ! | |NO ! |
| L | createPair | External | ! | ● |NO ! |
|||||
| **IV2Pair** | Interface | |||
| L | factory | External | ! | |NO ! |
| L | getReserves | External | ! | |NO ! |
| L | sync | External | ! | ● |NO ! |
|||||
| **IRouter01** | Interface | |||
| L | factory | External | ! | |NO ! |
| L | WETH | External | ! | |NO ! |
| L | addLiquidityETH | External | ! | 💰 |NO ! |
| L | addLiquidity | External | ! | ● |NO ! |
| L | swapExactETHForTokens | External | ! | 💰 |NO ! |
| L | getAmountsOut | External | ! | |NO ! |
| L | getAmountsIn | External | ! | |NO ! |
|||||
| **IRouter02** | Interface | IRouter01 |||
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ! | ● |NO ! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External | ! | 💰 |NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ! | ● |NO ! |
| L | swapExactTokensForTokens | External | ! | ● |NO ! |
|||||
| **IERC20** | Interface | |||
| L | totalSupply | External | ! | |NO ! |
| L | balanceOf | External | ! | |NO ! |
| L | allowance | External | ! | |NO ! |
| L | transfer | External | ! | ● |NO ! |
| L | approve | External | ! | ● |NO ! |
| L | transferFrom | External | ! | ● |NO ! |
|||||
| **IERC20Metadata** | Interface | IERC20 |||
| L | name | External | ! | |NO ! |
| L | symbol | External | ! | |NO ! |
| L | decimals | External | ! | |NO ! |
|||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| L | <Constructor> | Public | ! | ● |NO ! |
| L | name | Public | ! | |NO ! |
| L | symbol | Public | ! | |NO ! |
| L | decimals | Public | ! | |NO ! |
```

CONTRACT ASSESMENT

```

└─ totalSupply | Public ! | |NO ! |
└─ balanceOf | Public ! | |NO ! |
└─ transfer | Public ! | ● |NO ! |
└─ allowance | Public ! | |NO ! |
└─ approve | Public ! | ● |NO ! |
└─ transferFrom | Public ! | ● |NO ! |
└─ increaseAllowance | Public ! | ● |NO ! |
└─ decreaseAllowance | Public ! | ● |NO ! |
└─ _transfer | Internal 🔒 | ● | |
└─ _mint | Internal 🔒 | ● | |
└─ _burn | Internal 🔒 | ● | |
└─ _approve | Internal 🔒 | ● | |
└─ _beforeTokenTransfer | Internal 🔒 | ● | |
|||||
**DividendPayingTokenInterface** | Interface | |||
└─ dividendOf | External ! | |NO ! |
└─ withdrawDividend | External ! | ● |NO ! |
|||||
**DividendPayingTokenOptionalInterface** | Interface | |||
└─ withdrawableDividendOf | External ! | |NO ! |
└─ withdrawnDividendOf | External ! | |NO ! |
└─ accumulativeDividendOf | External ! | |NO ! |
|||||
**DividendPayingToken** | Implementation | ERC20, Ownable, DividendPayingTokenInterface,
DividendPayingTokenOptionalInterface |||
└─ <Constructor> | Public ! | ● | ERC20 |
└─ distributeDividends | Public ! | ● | onlyOwner |
└─ withdrawDividend | Public ! | ● |NO ! |
└─ _withdrawDividendOfUser | Internal 🔒 | ● | |
└─ dividendOf | Public ! | |NO ! |
└─ withdrawableDividendOf | Public ! | |NO ! |
└─ withdrawnDividendOf | Public ! | |NO ! |
└─ accumulativeDividendOf | Public ! | |NO ! |
└─ _transfer | Internal 🔒 | ● | |
└─ _mint | Internal 🔒 | ● | |
└─ _burn | Internal 🔒 | ● | |
└─ _setBalance | Internal 🔒 | ● | |
|||||
**DividendTracker** | Implementation | Ownable, DividendPayingToken |||
└─ <Constructor> | Public ! | ● | DividendPayingToken |
└─ _transfer | Internal 🔒 | | |
└─ withdrawDividend | Public ! | |NO ! |
└─ updateMinimumTokenBalanceForDividends | External ! | ● | onlyOwner |

```

CONTRACT ASSESMENT

```

└─ excludeFromDividends | External ! | ● | onlyOwner |
└─ updateClaimWait | External ! | ● | onlyOwner |
└─ setLastProcessedIndex | External ! | ● | onlyOwner |
└─ getLastProcessedIndex | External ! | |NO ! |
└─ getNumberOfTokenHolders | External ! | |NO ! |
└─ getAccount | Public ! | |NO ! |
└─ getAccountAtIndex | Public ! | |NO ! |
└─ canAutoClaim | Private 🔒 | | |
└─ setBalance | External ! | ● | onlyOwner |
└─ process | Public ! | ● |NO ! |
└─ processAccount | Public ! | ● | onlyOwner |
|||||
**TopGCEO** | Implementation | ERC20, Ownable |||
└─ <Constructor> | Public ! | 💰 | ERC20 |
└─ <Receive Ether> | External ! | 💰 |NO ! |
└─ changeWallets | External ! | ● | onlyOwner |
└─ excludeFromFees | External ! | ● | onlyOwner |
└─ isExcludedFromFees | Public ! | |NO ! |
└─ is_buy | Internal 🔒 | | |
└─ is_sell | Internal 🔒 | | |
└─ is_transfer | Internal 🔒 | | |
└─ _transfer | Internal 🔒 | ● | |
└─ swapAndLiquify | Internal 🔒 | ● | |
└─ takeTaxes | Internal 🔒 | ● | |
└─ swapAndSendDividends | Private 🔒 | ● | |
└─ setSwapTokensAtAmount | External ! | ● | onlyOwner |
└─ updateClaimWait | External ! | ● | onlyOwner |
└─ getClaimWait | External ! | |NO ! |
└─ getTotalDividendsDistributed | External ! | |NO ! |
└─ withdrawableDividendOf | Public ! | |NO ! |
└─ dividendTokenBalanceOf | Public ! | |NO ! |
└─ totalRewardsEarned | Public ! | |NO ! |
└─ excludeFromDividends | External ! | ● | onlyOwner |
└─ getAccountDividendsInfo | External ! | |NO ! |
└─ getAccountDividendsInfoAtIndex | External ! | |NO ! |
└─ processDividendTracker | External ! | ● |NO ! |
└─ claim | External ! | ● |NO ! |
└─ claimAddress | External ! | ● | onlyOwner |
└─ getLastProcessedIndex | External ! | |NO ! |
└─ setLastProcessedIndex | External ! | ● | onlyOwner |
└─ getNumberOfDividendTokenHolders | External ! | |NO ! |

```



CONTRACT ASSESMENT

Legend

Symbol	Meaning
:	:
●	Function can modify state
💰	Function is payable



STATIC ANALYSIS

```
DividendPayingToken.transfer(address,address,uint256) (contracts/Token.sol#888-891) is never used and should be removed
IterableMapping.get(IterableMapping.Map,address) (contracts/Token.sol#187-189) is never used and should be removed
SafeMath.div(uint256,uint256) (contracts/Token.sol#99-101) is never used and should be removed
SafeMath.div(uint256,uint256,string) (contracts/Token.sol#103-113) is never used and should be removed
SafeMath.mod(uint256,uint256) (contracts/Token.sol#115-117) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (contracts/Token.sol#119-126) is never used and should be removed
SafeMathInt.abs(int256) (contracts/Token.sol#158-161) is never used and should be removed
SafeMathInt.div(int256,int256) (contracts/Token.sol#141-144) is never used and should be removed
SafeMathInt.mul(int256,int256) (contracts/Token.sol#134-139) is never used and should be removed
TopGCEO.is_transfer(address,address) (contracts/Token.sol#1086-1092) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in TopGCEO.transfer(address,address,uint256) (contracts/Token.sol#1094-1199):
- (success,None) = marketingAddress.call{gas: 35000,value: mkt}() (contracts/Token.sol#1144-1146)
- (success,None) = charityAddress.call{gas: 35000,value: charity}() (contracts/Token.sol#1149-1152)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter01.WETH() (contracts/Token.sol#276) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address).owner (contracts/Token.sol#639) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address).owner (contracts/Token.sol#644) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address).owner (contracts/Token.sol#650) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address).owner (contracts/Token.sol#656) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#579) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter DividendTracker.updateMinimumTokenBalanceForDividends(uint256).newMinimumBalance (contracts/Token.sol#755) is not in mixedCase
Parameter DividendTracker.getAccount(address).account (contracts/Token.sol#800) is not in mixedCase
Event TopGCEODividendRevert() (contracts/Token.sol#1257) is not in CapWords
Function TopGCEO.is_buy(address,address) (contracts/Token.sol#1076-1079) is not in mixedCase
Function TopGCEO.is_sell(address,address) (contracts/Token.sol#1081-1084) is not in mixedCase
Function TopGCEO.is_transfer(address,address) (contracts/Token.sol#1086-1092) is not in mixedCase
Constant TopGCEO.buyfee (contracts/Token.sol#974) is not in UPPER_CASE_WITH_UNDERSCORES
Constant TopGCEO.sellfee (contracts/Token.sol#975) is not in UPPER_CASE_WITH_UNDERSCORES
Constant TopGCEO.transferfee (contracts/Token.sol#976) is not in UPPER_CASE_WITH_UNDERSCORES
Constant TopGCEO.fee_denominator (contracts/Token.sol#977) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#18)" inContext (contracts/Token.sol#10-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#293) is too similar to IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#294)
Variable DividendPayingToken.withdrawDividendOfUser(address).withdrawableDividend (contracts/Token.sol#616) is too similar to DividendTracker.getAccount(address).withdrawableDividends (contracts/Token.sol#808)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SafeMathInt.MAX_INT256 (contracts/Token.sol#132) is never used in SafeMathInt (contracts/Token.sol#130-167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

TopGCEO.gasForProcessing (contracts/Token.sol#993) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

TopGCEO.dividendTracker (contracts/Token.sol#991) should be immutable
TopGCEO.lpPair (contracts/Token.sol#978) should be immutable
TopGCEO.swapRouter (contracts/Token.sol#985) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Static Analysis

an static analysis of the code were performed using slither. No issues were found



FUNCTIONAL TESTING

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0x75226ddda3c47de3ae71c9128cc9e4d82b6d809df050cdfae441b1247cbae172>

2- Buying when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xc418f5a9df0f9efccc8cb4424b34de7307822d2a6f4dad22529cd127d73e90ef>

3- Selling when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xfbe61d8f1dc6b37de067a31ecc09ac3e2c1493e5b7f6bde5e3f6a31815203146>

4- Transferring when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xfbe61d8f1dc6b37de067a31ecc09ac3e2c1493e5b7f6bde5e3f6a31815203146>

5- Buying when not excluded from fees (8% tax) (passed):

<https://testnet.bscscan.com/tx/0xddbae25930ef11ad93a897d1968086a010e03798ecc89da10d72b808998f5253>

6- Selling when not excluded from fees (8% tax) (passed):

<https://testnet.bscscan.com/tx/0xbdb9d3a7a2dd0624ac89cf12b2cfc27748a6e751659fd2638cc742f564994b10>

7- Transferring when not excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x781ac5cf602e61200a694fa16185033cee5b6154dfacf777af55330844788a06>



FUNCTIONAL TESTING

8- Internal swap (marketing and charity bnb + BT CB distribution + auto liquidity) (passed):

<https://testnet.bscscan.com/tx/0xbdb9d3a7a2dd0624ac89cf12b2cfc27748a6e751659fd2638cc742f564994b10>

FUNCTIONAL TESTING

Logical – Excessive swap threshold

Severity: **Informational**

function: setSwapTokensAtAmount

Status: Not Applicable (Resolved)

Overview:

Owner is able to set swapTokensAtAmount to a very large value which can potentially lead to increased slippage and risk of being front-run on trades

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount >= totalSupply() / 1_000_000,
        "SwapTokensAtAmount must be greater than 0.001% of total supply"
    );
    swapTokensAtAmount = amount;
}
```

Suggestion

To mitigate this centralization issue, ensure that swap threshold is less than a reasonable maximum value, depending on liquidity pool size this maximum value can be variant, but its a common practice to consider 1% of total supply as maximum value of swap threshold.

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount >= totalSupply() / 1_000_000,
        "SwapTokensAtAmount must be greater than 0.001% of total supply"
    );
    require(
        amount <= totalSupply() / 1_00,
        "SwapTokensAtAmount must be less than 1% of total supply"
    );
    swapTokensAtAmount = amount;
}
```



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 specialize 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>
