AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

BETCOIN CASH

DATED : 28 JAN 23'

AUDIT SUMMARY

Project name – BETCOIN CASH

Date: 28 January , 2023

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

Audit Status: Passed (Contract is developed by Pinksale safu dev)

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	1
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 done on BSC Testnet network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0xc45b6ca0c252
8c71f863f63127e6ff2badb5fdf6

Token Information

Token Name : Betcoin Cash

Token Symbol: BETC

Decimals: 9

Token Address:
0xdf57a68dc986924a09469dD7AFA650A5E5A3d32C

Checksum:
89328eb1ccf7bd0b115ecd91f95d7136fd5c3e571db1c3
37e7bd356c8bfec0f3

Deployer:
0xcb64678C574C1e3A900a0010bC98eF2f79099C57

Owner:
0xcb64678C574C1e3A900a0010bC98eF2f79099C57

TOKEN OVERVIEW

Fees:

Buy Fees: - 0%

Sell Fees: - 20%

Transfer Fees: - 0%

Fees Privilige: Owner

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: None

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

- ✅ Return values of low-level calls
- ✅ Private modifier
- ✅ Multiple Sends
- ✅ Using Suicide
- ✅ Gas Limitand Loops
- ✅ Address hardcoded
- ✅ Exception Disorder
- ✅ Using inline assembly
- ✅ Divide before multiply
- ✅ Missing Zero Address Validation
- ✅ Compiler version not fixed

- ✅ **Gasless Send**
- ✅ Using block.timestamp
- ✅ Re-entrancy
- ✅ Tautology or contradiction
- ✅ Timestamp Dependence
- ✅ Revert/require functions
- ✅ Use of tx.origin
- ✅ Integer overflow/underflow
- ✅ Dangerous strict equalities
- ✅ Using SHA3
- ✅ 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

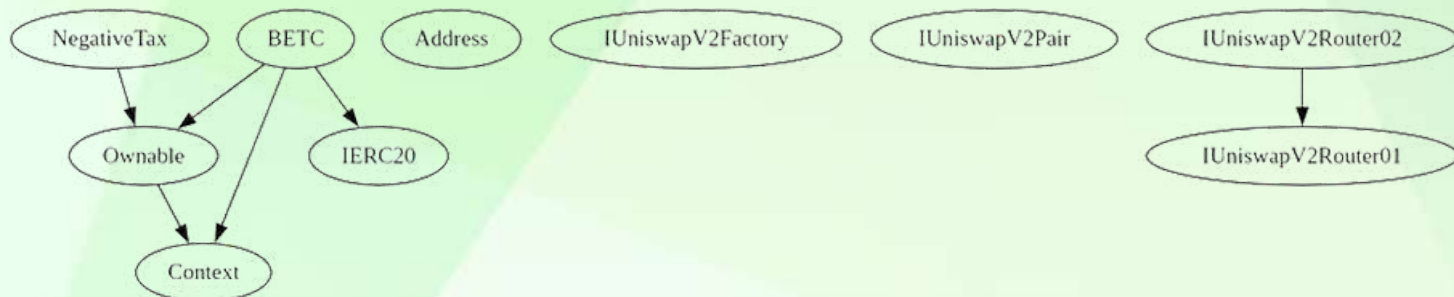| Severity	Found
◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	1

INHERITANCE TREE

POINTS TO NOTE

- **There is up to 5% cashback on buys (negative tax)**

- **Owner is not able to set taxes over 25% (Buy + Sell = 50% maximum tax)**

- **Owner is not able to blacklist an arbitrary wallet**

- **Owner is not able to set max buy/sell/transfer amounts**

- **Owner is not able to disable trades**

- **Owner is not able to mint new tokens**

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Pair	Interface			

CONTRACT ASSESMENT

└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗

CONTRACT ASSESMENT

└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
NegativeTax	Implementation	Ownable		
└	claimStuckTokens	External ❗	🛑	onlyOwner
BETC	Implementation	Context, IERC20, Ownable		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	isExcludedFromReward	Public ❗		NO❗
└	totalReflectionDistributed	Public ❗		NO❗
└	deliver	Public ❗	🛑	NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO❗
└	excludeFromNegativeTax	External ❗	🛑	onlyOwner
└	isExcludedFromNegativeTax	Public ❗		NO❗
└	setNegativeTaxSettings	External ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	claimStuckTokens	External ❗	🛑	onlyOwner
└	_reflectFee	Private 🔒	🛑	
└	_getValues	Private 🔒		
└	_getTValues	Private 🔒		
└	_getRValues	Private 🔒		
└	_getRate	Private 🔒		
└	_getCurrentSupply	Private 🔒		
└	_takeLiquidity	Private 🔒	🛑	
└	_takeMarketing	Private 🔒	🛑	
└	calculateTaxFee	Private 🔒		
└	calculateLiquidityFee	Private 🔒		
└	calculateMarketingFee	Private 🔒		
└	removeAllFee	Private 🔒	🛑	
└	setBuyFee	Private 🔒	🛑	
└	setSellFee	Private 🔒	🛑	
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔒	🛑	
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Private 🔒	🛑	
└	swapAndLiquify	Private 🔒	🛑	
└	swapAndSendMarketing	Private 🔒	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔒	🛑	
└	_transferStandard	Private 🔒	🛑	
└	_transferToExcluded	Private 🔒	🛑	
└	_transferFromExcluded	Private 🔒	🛑	
└	_transferBothExcluded	Private 🔒	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	setBuyFeePercentages	External ❗	🛑	onlyOwner
└	setSellFeePercentages	External ❗	🛑	onlyOwner
└	enableWalletToWalletTransferWithoutFee	External ❗	🛑	onlyOwner

Symbol	Meaning
🛑	Function can modify state
🏧	Function is payable

STATIC ANALYSIS

```
Variable BETC._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1238) is too similar to BETC._getTValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#917)
Variable BETC.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Ace_Testing_BSC.sol#795) is too similar to BETC._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1286)
Variable BETC._getValues(uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#893) is too similar to BETC._getValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#888)
Variable BETC._getValues(uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#893) is too similar to BETC._getTValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#917)
Variable BETC.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Ace_Testing_BSC.sol#795) is too similar to BETC._getTValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#917)
Variable BETC._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1261) is too similar to BETC._getValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#888)
Variable BETC._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1261) is too similar to BETC._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1286)
Variable BETC._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1261) is too similar to BETC._getTValues(uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#917)
Variable BETC._transferStandard(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1216) is too similar to BETC._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1240)
Variable BETC._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#932) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Variable BETC.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Ace_Testing_BSC.sol#795) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Variable BETC._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1261) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Variable BETC._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1284) is too similar to BETC._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1240)
Variable BETC._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1284) is too similar to BETC._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1263)
Variable BETC._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#932) is too similar to BETC._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1240)
Variable BETC.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Ace_Testing_BSC.sol#795) is too similar to BETC._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1240)
Variable BETC._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1238) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Variable BETC._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1261) is too similar to BETC._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1240)
Variable BETC._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#1284) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Variable BETC._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#932) is too similar to BETC._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1263)
Variable BETC._getValues(uint256).rTransferAmount (contracts/Ace_Testing_BSC.sol#893) is too similar to BETC._transferStandard(address,address,uint256).tTransferAmount (contracts/Ace_Testing_BSC.sol#1218)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

BETC._decimals (contracts/Ace_Testing_BSC.sol#556) should be constant
BETC._name (contracts/Ace_Testing_BSC.sol#554) should be constant
BETC._symbol (contracts/Ace_Testing_BSC.sol#555) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

BETC._tTotal (contracts/Ace_Testing_BSC.sol#559) should be immutable
BETC.negativeTax (contracts/Ace_Testing_BSC.sol#563) should be immutable
BETC.uniswapV2Pair (contracts/Ace_Testing_BSC.sol#592) should be immutable
BETC.uniswapV2Router (contracts/Ace_Testing_BSC.sol#591) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither, no major issues were found (except some "minor impact" suggestions)

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

0- Deploying (Passed):

https://testnet.bscscan.com/tx/0x83d209b9e868916315a9c15cfcb9ddd7a47c11f0943559588e782e8ad3453d27

1- Adding Liquidity (Passed):
liquidity added on Pancakeswap V2:

https://testnet.bscscan.com/tx/0x15443cc8182367ed3da4de488a8c7d2a832e76d767a6c87834f01eba3b539a95

no issue were found on adding liquidity.

2- Enabling trade for public (Passed):

https://testnet.bscscan.com/tx/0x88d8f6898921ab9b08105049d3410a098c64aaa63d1a8a129da6a8238b1b4bbd

3- Excluding deployer's wallet from taxes (to test taxes) (Passed):

https://testnet.bscscan.com/tx/0x8a2d2f1369358c0ab93d6cd07036a51b20d34fe2aa17d0a674b60365d89387b1

FUNCTIONAL TESTING

4- Enabling swap and liquify (Passed):

https://testnet.bscscan.com/tx/0x8a65708bd50e73dde9e210d014799d28ac0229bef5c2aa77599a475d25de8349

5- Setting buy and sell fees to max (25%) (Passed):

https://testnet.bscscan.com/tx/0x076b6db648ee67c90e1bf93a33f8ad97c3339f60552c6844e9a73814d9d3ebd2

https://testnet.bscscan.com/tx/0x686788d3ebd5d80d2206454d0269d606910f475a3b06346b07e9dafce160a335

6- Buying from a non-whitelisted wallet (Passed):
https://testnet.bscscan.com/tx/0x7659a5a0760ffbbbf8fde411d4b3c4d26185f5c020f64dd027f20d37376823ef

cashback tx:
https://testnet.bscscan.com/address/0x890c6ae98ae2962f73d762d38416702843222dff#tokentxns
25% - 5% cashback => 20% tax applied

7- Selling from a non-whitelisted wallet (Passed):

https://testnet.bscscan.com/tx/0xdc05f5bafbad470f2704c6cb7736fb6eeeef127287950b7b02c6aca363ad9f7b

25% tax applied

FUNCTIONAL TESTING

8- **Internal Swap** (Passed):

taxes were swapped to BNB and sent to marketing wallet which can be seen here:

https://testnet.bscscan.com/address/0x024238ae95bf8e4361ba58ca018e84c12c654509#internaltx

9- **Auto Liquidity** (Passed):

LP tokens generated and sent to dead wallet which can be seen here:

https://testnet.bscscan.com/token/0x4cb202df73c90485bf4df97e7076a5275b31ce63?a=0x000000000000000000000000000000000000dead

10- **6- Enabling negative tax (cashback, up to 5%)** (Passed):

https://testnet.bscscan.com/tx/0xf539dc09cab6531c332db82816ec402b2a28a3979e8166a255bc24e6f21583a8

FUNCTIONAL TESTING

11- Transferring while transfer taxes are off (wallet to wallet transfer without fee) (Passed):

disabling transfer fees:

https://testnet.bscscan.com/tx/0xca3fdada83f0637cf6454a20251ec384b97d3c8959bcb0adbfa7e3a4b6a801e7

transferring 100,000,000 tokens from a non-whitelisted wallet to a non-whitelisted wallet without fee:

https://testnet.bscscan.com/tx/0xd4da6a451e6594af93ec22f17388e4fc2172dd7e65ecd2805e00aefd7989d856

MANUAL TESTING

Suggestions

- **Emit a transfer event at _takeMarketing and _takeLiquidity functions**

Social Media Overview

Here are the Social Media Accounts of

BetCoin

https://t.me/BetcoinCashOfficial

https://twitter.com/Betcoin_Cash

https://betcoincash.io/

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