auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

BitcoinETFToken

DATED : 14 Dec 23'

AUDIT SUMMARY

Project name – BitcoinETFToken

Date: 14 Dec, 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	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 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/0xc2ad81168cc9bb3812b1c62c34d3d9c036cac311#code

Token Information

Token Address:
0x0d825acA6c9D4e1595254C419138bbaC2d4D7abB

Symbol: BTCETF

Decimals: 18

Network: BscScan

Token Type: BEP-20

Owner: 0x853ffB780c52a23baFaaC23bb6Ffe2ec9f46635A

Deployer: 0x853ffB780c52a23baFaaC23bb6Ffe2ec9f46635A

Token Supply: 100000000000000000000000000

Checksum: Ef126dac9919ad76433d7e81ee6d9a43

Testnet:
https://testnet.bscscan.com/address/0xc2ad81168cc9bb3812b1c62c34d3d9c036cac311#code

TOKEN OVERVIEW

Buy Fee: 0-5%

Sell Fee: 0-5%

Transfer Fee: 0-0%

Fee Privilege: Owner

Ownership: Owned

Minting: None

Max Tx: Yes

Blacklist: No

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

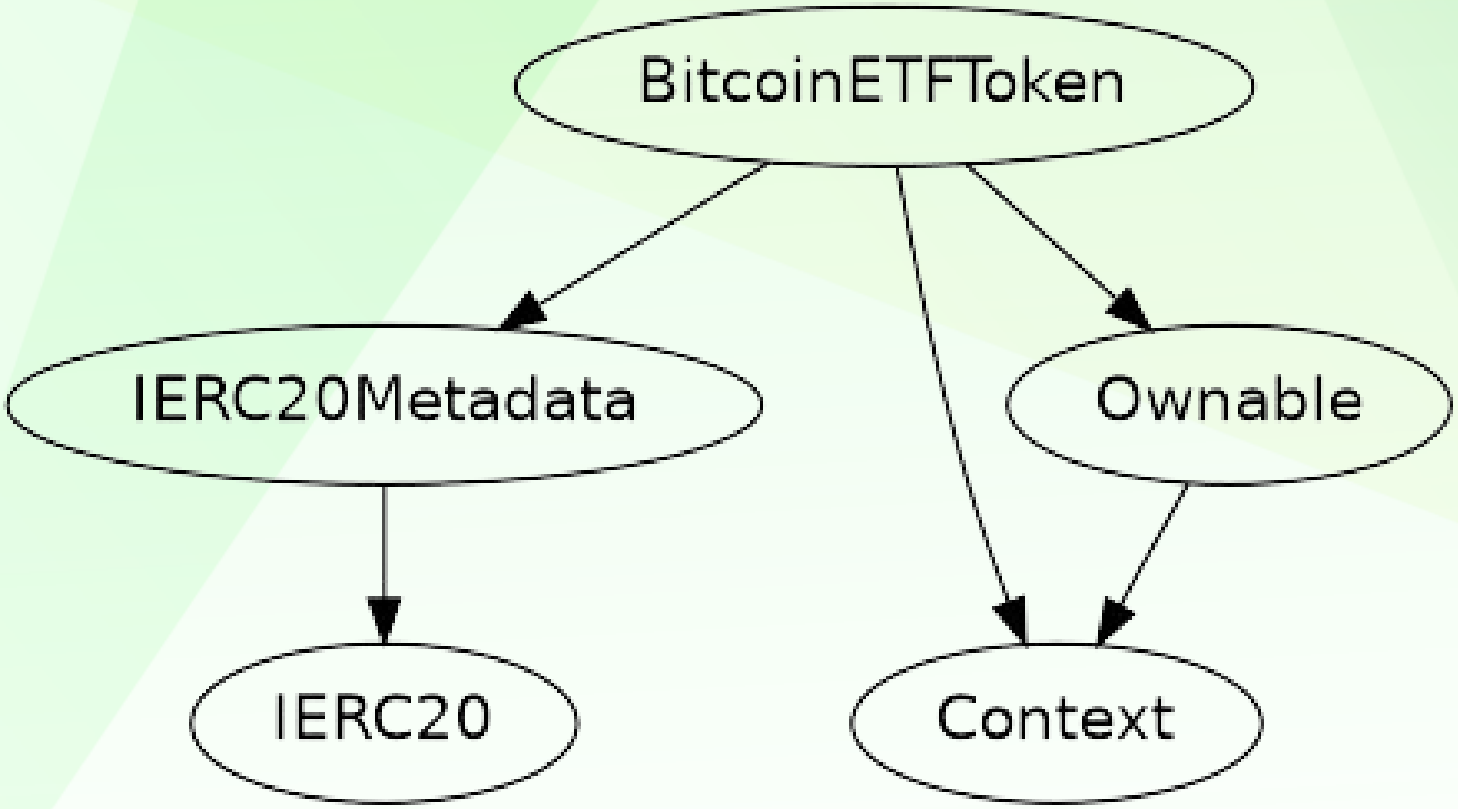
CLASSIFICATION OF RISK

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

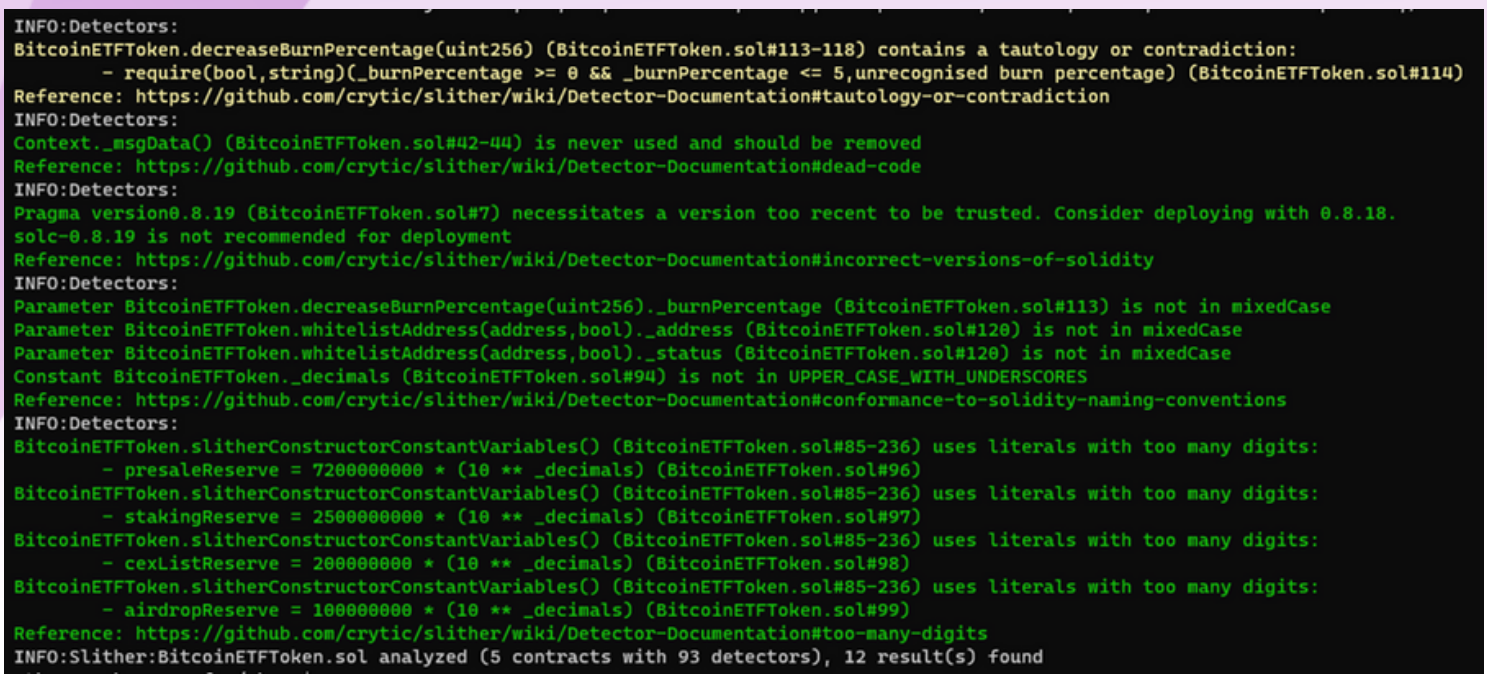
INHERITANCE TREE



POINTS TO NOTE

- The owner can renounce the ownership.
- The owner can transfer ownership.
- The owner can change the marketing wallet address.
- The owner can set buy and sell fees of not more than 5%.
- The owner can Include/exclude the address from fees.

STATIC ANALYSIS



```
INFO:Detectors:
BitcoinETFToken.decreaseBurnPercentage(uint256) (BitcoinETFToken.sol#113-118) contains a tautology or contradiction:
        - require(bool,string)(_burnPercentage >= 0 && _burnPercentage <= 5,unrecognised burn percentage) (BitcoinETFToken.sol#114)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction
INFO:Detectors:
Context._msgData() (BitcoinETFToken.sol#42-44) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.19 (BitcoinETFToken.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.8.18.
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter BitcoinETFToken.decreaseBurnPercentage(uint256)._burnPercentage (BitcoinETFToken.sol#113) is not in mixedCase
Parameter BitcoinETFToken.whitelistAddress(address,bool)._address (BitcoinETFToken.sol#120) is not in mixedCase
Parameter BitcoinETFToken.whitelistAddress(address,bool)._status (BitcoinETFToken.sol#120) is not in mixedCase
Constant BitcoinETFToken._decimals (BitcoinETFToken.sol#94) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
BitcoinETFToken.slitherConstructorConstantVariables() (BitcoinETFToken.sol#85-236) uses literals with too many digits:
        - presaleReserve = 7200000000 * (10 ** _decimals) (BitcoinETFToken.sol#96)
BitcoinETFToken.slitherConstructorConstantVariables() (BitcoinETFToken.sol#85-236) uses literals with too many digits:
        - stakingReserve = 2500000000 * (10 ** _decimals) (BitcoinETFToken.sol#97)
BitcoinETFToken.slitherConstructorConstantVariables() (BitcoinETFToken.sol#85-236) uses literals with too many digits:
        - cexListReserve = 200000000 * (10 ** _decimals) (BitcoinETFToken.sol#98)
BitcoinETFToken.slitherConstructorConstantVariables() (BitcoinETFToken.sol#85-236) uses literals with too many digits:
        - airdropReserve = 100000000 * (10 ** _decimals) (BitcoinETFToken.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Slither:BitcoinETFToken.sol analyzed (5 contracts with 93 detectors), 12 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

FUNCTIONAL TESTING

1- Approve (passed):

https://testnet.bscscan.com/tx/0x895977d7bb12c8c565687fb898ac016c9daf6aaa415bd115e11c758bf4c48215

2- Increase Allowance (passed):

https://testnet.bscscan.com/tx/0x302a237d08abe354f200908d2606ef3d92a3778b682561f74100da04cff16d57

3- Decrease Allowance (passed):

https://testnet.bscscan.com/tx/0x7c3ca22b6218d686f106e1bdf3c50c0c6fe933b29f66191e62b06f6801fbcc40

4- Decrease Burn Percentage (passed):

https://testnet.bscscan.com/tx/0x4808af26e0f2c57b0903d991d3d9dad0e2283dc85caee07f320cd09809bd665e

5- Whitelist Address (passed):

https://testnet.bscscan.com/tx/0x03f468760c2cc4115ca1543361e8623a32322fa1f262fd2a4a9a47e0dffadf11

MANUAL TESTING

Optimization

Severity: Informational

subject: Remove unused code.

Status: Open

Overview:

Unused variables are allowed in Solidity, and they do. not pose a direct security issue. It is the best practice. though to avoid them.

```
function _msgData() internal view virtual returns (bytes calldata) {
  return msg.data;
 }
```

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