auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

TRUMP47

DATED : 13 Feb, 2024

MANUAL TESTING

Centralization – Missing Require Check

Severity: High

Function: SetMarketingWallet

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner will set the address to the contract address, then the Eth will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMarketinngWallet(address newWallet) external onlyOwner() {
        require(newWallet != address(0),
            "BEP20: the new wallet cannot be the zero address."
        );
        marketingWallet = newWallet;
    }
```

Suggestion:

It is recommended that the address should not be able to set as a contract address.

AUDIT SUMMARY

Project name – TRUMP47

Date: 13 Feb, 2024

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

Audit Status: **Passed With High Risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	0	2	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/0x39F17AC0a63f633FC6ea3bBd7ADA08AE7E2E9856#code

Token Information

Token Name : Trump47thepresident

Token Symbol: Trump47

Decimals: 18

Token Supply: 100000000000

Network: BscScan

Token Type: BEP-20

Token Address:
0xeC8A3F9e1cA1A884b68e4d1dc13b42E584D03236

Checksum:
H2032c616934aeb47e6039f76b20d212

Owner:
0xB56377b51Bd3998b25Ce5C21cB83710735be9640
(at time of writing the audit)

Deployer:
0x8EEDF105fb98054833F2fa8F77c13840Fb29177B

TOKEN OVERVIEW

Fees:

Marketing Tax: 3%

Donation Fee: 1%

Tax Fee: 4%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Note: The owner can set the max tax amount but not more than the max tax amount which is mentioned in the contract i.e. 100000000000

AUDIT METHODOLOGY

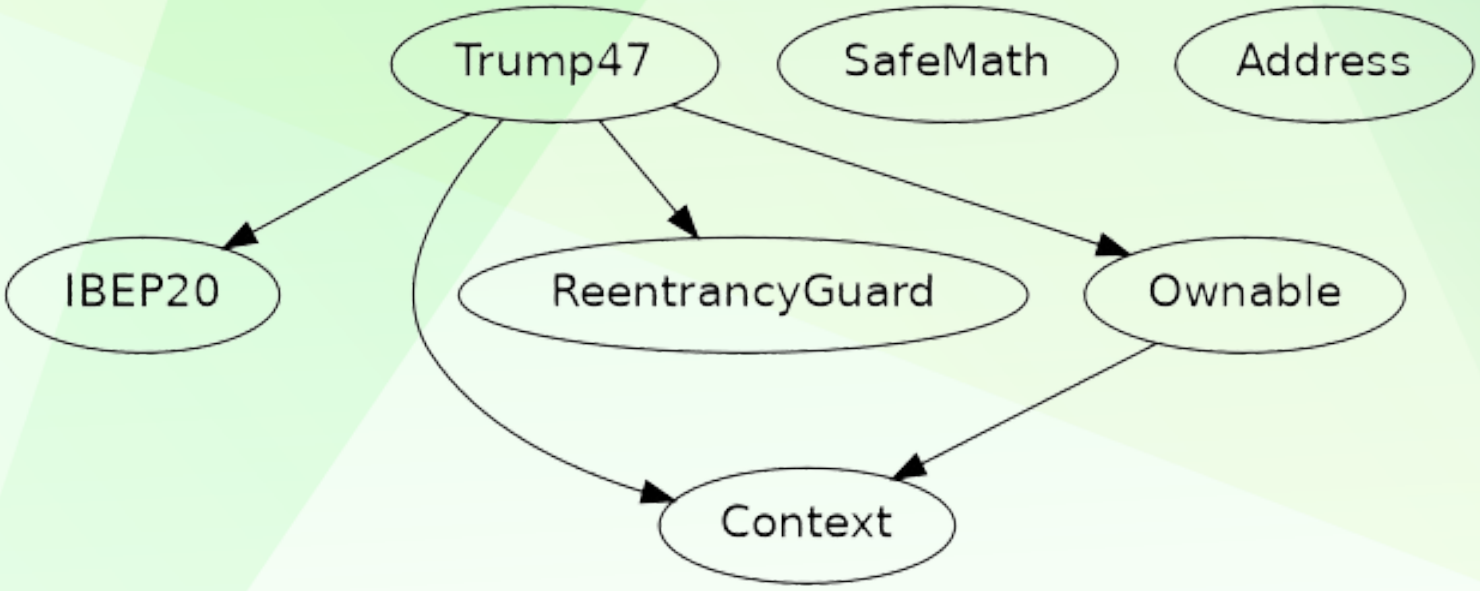
The auditing process will follow a routine as special considerations by Auditace:

- 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

INHERITANCE TREE



STATIC ANALYSIS

A static analysis of the code was performed using Slither.

No issues were found.

```
INFO:Detectors:
Trump47.allowance(address,address).owner (Trump.sol#779) shadows:
        - Ownable.owner() (Trump.sol#626-628) (function)
Trump47._approve(address,address,uint256).owner (Trump.sol#1072) shadows:
        - Ownable.owner() (Trump.sol#626-628) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Trump47.setMaxTxAmount(uint256) (Trump.sol#1273-1278) should emit an event for:
        - _maxTxAmount = newAmount (Trump.sol#1277)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Address.isContract(address) (Trump.sol#366-376) uses assembly
        - INLINE ASM (Trump.sol#374)
Address._functionCallWithValue(address,bytes,uint256,string) (Trump.sol#485-511) uses assembly
        - INLINE ASM (Trump.sol#503-506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Trump47.includeInReward(address) (Trump.sol#917-928) has costly operations inside a loop:
        - _excluded.pop() (Trump.sol#924)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Address._functionCallWithValue(address,bytes,uint256,string) (Trump.sol#485-511) is never used and should be removed
Address.functionCall(address,bytes) (Trump.sol#423-428) is never used and should be removed
Address.functionCall(address,bytes,string) (Trump.sol#436-442) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Trump.sol#455-466) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (Trump.sol#475-483) is never used and should be removed
Address.isContract(address) (Trump.sol#366-376) is never used and should be removed
Address.sendValue(address,uint256) (Trump.sol#396-402) is never used and should be removed
Context._msgData() (Trump.sol#528-532) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Trump.sol#306-315) is never used and should be removed
SafeMath.mod(uint256,uint256) (Trump.sol#266-268) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Trump.sol#332-341) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (Trump.sol#135-141) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (Trump.sol#178-183) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (Trump.sol#191-196) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (Trump.sol#160-170) is never used and should be removed
SafeMath.trySub(uint256,uint256) (Trump.sol#148-153) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Trump47._previousTaxFee (Trump.sol#696) is set pre-construction with a non-constant function or state variable:
        - _taxFee
Trump47._previousDonationFee (Trump.sol#703) is set pre-construction with a non-constant function or state variable:
        - _donationFee
Trump47._previousMarketingFee (Trump.sol#710) is set pre-construction with a non-constant function or state variable:
        - _marketingFee
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state
```

STATIC ANALYSIS

FUNCTIONAL TESTING

1- **Approve (passed):**
https://testnet.bscscan.com/tx/0x29a6c4dcf8b6dd9df5fa3ae3a7b9f790c4b2d4a9a743004409d98c6acb2e1839

2- Enable All Fees (passed):
https://testnet.bscscan.com/tx/0xb9ee064d8d55f4e03bf06baabae031c80d1bbefc05e0479818a82a879d8070d0

3- Disable All Fees (passed):
https://testnet.bscscan.com/tx/0xf6e1ba7170cf3e4e0c2e5d8322f37b858cd0b67e24a3b66e67ad2a8d61c9de03

4- Exclude From Fee (passed):
https://testnet.bscscan.com/tx/0xc181f5e83852ed8b1e1067982f7b9f7b9cddbe64e18aae5eaa72b40098bf4a0b

5- Include In Fee (passed):
https://testnet.bscscan.com/tx/0x7dd3dd333f98c994853280b293142a631642bb938cdad322ead346d8225af045

6- Set Marketing Wallet (passed):
https://testnet.bscscan.com/tx/0x824e67a855c3db18e80f1cddee8b586fea086b71309cab22acd3baad8e4d0a17

POINTS TO NOTE

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can exclude/include address from rewards.
- The owner can exclude/include address from fee.
- The owner can enable/disable all fees.
- The owner can stop donation fee.
- The owner can set marketing wallet addresses.

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	1
Medium-Risk	0
Low-Risk	2
Gas Optimization / Suggestions	1

MANUAL TESTING

Centralization – Missing Require Check

Severity: High

Function: SetMarketingWallet

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner will set the address to the contract address, then the Eth will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMarketinngWallet(address newWallet) external onlyOwner() {
        require(newWallet != address(0),
            "BEP20: the new wallet cannot be the zero address."
        );
        marketingWallet = newWallet;
    }
```

Suggestion:

It is recommended that the address should not be able to set as a contract address.

MANUAL TESTING

Centralization – Missing Events

Severity: Low

Subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setMarketinngWallet(address newWallet) external onlyOwner() {
        require(newWallet != address(0),
            "BEP20: the new wallet cannot be the zero address."
        );
        marketingWallet = newWallet;
    }
function setMaxTxAmount(uint256 newAmount) external onlyOwner() {
        require(newAmount >= 1e6 * 10**_DECIMALS,
            "BEP20: the amount must be greater than or equal to one million."
        );
        _maxTxAmount = newAmount;
    }
function enableAllFees() external onlyOwner() {
        _taxFee = 4;
        _previousTaxFee = _taxFee;
        _donationFee = 1;
        _previousDonationFee = _donationFee;
        _marketingFee = 3;
        _previousMarketingFee = _marketingFee;
    }
function disableAllFees() external onlyOwner() {
        _taxFee = 0;
        _previousTaxFee = _taxFee;
        _donationFee = 0;
        _previousDonationFee = _donationFee;
        _marketingFee = 0;
        _previousMarketingFee = _marketingFee;
    }
```

MANUAL TESTING

Centralization – Local variable Shadowing

Severity: Low

Subject: Variable Shadowing

Status: Open

Overview:

```
function _approve(address owner, address spender, uint256 amount) private {
        require(owner != address(0), "BEP20: approve from the zero address");
        require(spender != address(0), "BEP20: approve to the zero address");

        _allowances[owner][spender] = amount;
        emit Approval(owner, spender, amount);
    }
function allowance(address owner, address spender)
        external
        view
        override
        returns (uint256)
    {
        return _allowances[owner][spender];
    }
```

Suggestion:

Rename the local variables that shadow another component.

MANUAL TESTING

Optimization

Severity: Informational

Subject: Remove Safe Math

Status: Open

Line: 129-342

Overview:

compiler version above 0.8.0 can control arithmetic overflow/underflow, it is recommended to remove the unwanted code to avoid high gas fees.

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