



Cyberscope

Audit Report

ETHnothing

June 2023

Github <https://github.com/ETHnothing/smart-contracts>

Commit [0fd0bd143a67d8d46e04ca10e1ddc641e88acbea](#)

Audited by © cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

| Severity | Code | Description | Status |
|----------|------|-------------------------|--------|
| ● | ST | Stops Transactions | Passed |
| ● | OTUT | Transfers User's Tokens | Passed |
| ● | ELFM | Exceeds Fees Limit | Passed |
| ● | MT | Mints Tokens | Passed |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Passed |

Diagnostics

● Critical ● Medium ● Minor / Informative

| Severity | Code | Description | Status |
|----------|------|--|------------|
| ● | PAV | Pair Address Validation | Unresolved |
| ● | EFLC | Exclude Function Logic Concern | Unresolved |
| ● | L02 | State Variables could be Declared Constant | Unresolved |

Table of Contents

| | |
|--|-----------|
| Analysis | 1 |
| Diagnostics | 2 |
| Table of Contents | 3 |
| Review | 4 |
| Audit Updates | 4 |
| Source Files | 5 |
| Findings Breakdown | 6 |
| PAV - Pair Address Validation | 7 |
| Description | 7 |
| Recommendation | 7 |
| EFLC - Exclude Function Logic Concern | 8 |
| Description | 8 |
| Recommendation | 8 |
| L02 - State Variables could be Declared Constant | 9 |
| Description | 9 |
| Recommendation | 9 |
| Functions Analysis | 10 |
| Inheritance Graph | 11 |
| Flow Graph | 12 |
| Summary | 13 |
| Disclaimer | 14 |
| About Cyberscope | 15 |

Review

| | |
|----------------|---|
| Contract Name | Nothing |
| Repository | https://github.com/ETHnothing/smart-contracts |
| Commit | 0fddbd143a67d8d46e04ca10e1ddc641e88acbea |
| Testing Deploy | https://testnet.bscscan.com/address/0x7fbb3d026d25333bee1f839570f62b09a1e02908 |
| Symbol | TST |
| Decimals | 8 |
| Total Supply | 1.000.000.000 |

Audit Updates

| | |
|-------------------|--|
| Initial Audit | 01 Jun 2023 https://github.com/cyberscope-io/audits/blob/main/ethnothing/v1/audit.pdf |
| Corrected Phase 2 | 22 Jun 2023 https://github.com/cyberscope-io/audits/blob/main/ethnothing/v2/audit.pdf |
| Corrected Phase 3 | 23 Jun 2023 |

Source Files

| Filename | SHA256 |
|---|--|
| @openzeppelin/contracts/access/Ownable.sol | 9353af89436556f7ba8abb3f37a6677249a a4df6024fbfaa94f79ab2f44f3231 |
| @openzeppelin/contracts/token/ERC20/ERC20.sol | bce14c3fd3b1a668529e375f6b70ffdf9cef 8c4e410ae99608be5964d98fa701 |
| @openzeppelin/contracts/token/ERC20/extensions /ERC20Burnable.sol | 0344809a1044e11ece2401b4f7288f414ea 41fa9d1dad24143c84b737c9fc02e |
| @openzeppelin/contracts/token/ERC20/extensions /IERC20Metadata.sol | af5c8a77965cc82c33b7ff844deb9826166 689e55dc037a7f2f790d057811990 |
| @openzeppelin/contracts/token/ERC20/IERC20.sol | 94f23e4af51a18c2269b355b8c7cf4db800 3d075c9c541019eb8dcf4122864d5 |
| @openzeppelin/contracts/utils/Context.sol | 1458c260d010a08e4c20a4a517882259a2 3a4baa0b5bd9add9fb6d6a1549814a |
| contracts/Nothing.sol | 48d16a244c877b9794ec326ba4eafb8daf7 cca9c37f3583adf6bd291b3991ff2 |

Findings Breakdown



| | |
|-----------------------|---|
| ● Critical | 0 |
| ● Medium | 0 |
| ● Minor / Informative | 3 |

| Severity | Unresolved | Acknowledged | Resolved | Other |
|-----------------------|------------|--------------|----------|-------|
| ● Critical | 0 | 0 | 0 | 0 |
| ● Medium | 0 | 0 | 0 | 0 |
| ● Minor / Informative | 3 | 0 | 0 | 0 |

PAV - Pair Address Validation

| | |
|-------------|---------------------------|
| Criticality | Minor / Informative |
| Location | contracts/Nothing.sol#L62 |
| Status | Unresolved |

Description

The contract is missing address validation in the pair address argument. The absence of validation reveals a potential vulnerability, as it lacks proper checks to ensure the integrity and validity of the pair address provided as an argument. The pair address is a parameter used in certain methods of decentralized exchanges for functions like token swaps and liquidity provisions.

The absence of address validation in the pair address argument can introduce security risks and potential attacks. Without proper validation, if the owner's address is compromised, the contract may lead to unexpected behavior like loss of funds.

```
function setMarketPair(address pair, bool state) public onlyOwner {
    require(pair != address(0), "Pair can't be address zero");
    if (marketPairs[pair] != state) {
        marketPairs[pair] = state;
        emit MarketPairUpdated(pair, state);
    }
}
```

Recommendation

To mitigate the risks associated with the absence of address validation in the pair address argument, it is recommended to implement comprehensive address validation mechanisms. A recommended approach could be to verify pair existence in the decentralized application. Prior to interacting with the pair address contract, perform checks to verify the existence and validity of the contract at the provided address. This can be achieved by querying the provider's contract or utilizing external libraries that provide contract verification services.

EFLC - Exclude Function Logic Concern

| | |
|-------------|----------------------------|
| Criticality | Minor / Informative |
| Location | contracts/Nothing.sol#L121 |
| Status | Unresolved |

Description

The contract implements an exclusion mechanism to exempt addresses from being charged fees. However, the implementation only checks if the sender `from` address is excluded from fees and not the recipient `to` too. This inconsistency can lead to unintended fee deductions for recipients who should be exempted from fees.

```
bool isExcludedFromFees = excludedFromFees[from];
if (!isExcludedFromFees) {
    // If to == marketPairs::true state, then it is a sell transfer
    if (marketPairs[to]) {
        taxAmount = (amount * SELL_FEE / 10_000);
    }
}
```

Recommendation

It is recommended to check the exclusion status of both recipient and sender addresses before deducting any fees.

L02 - State Variables could be Declared Constant

| | |
|-------------|---------------------------|
| Criticality | Minor / Informative |
| Location | contracts/Nothing.sol#L19 |
| Status | Unresolved |

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint8 private _decimals = 8
```

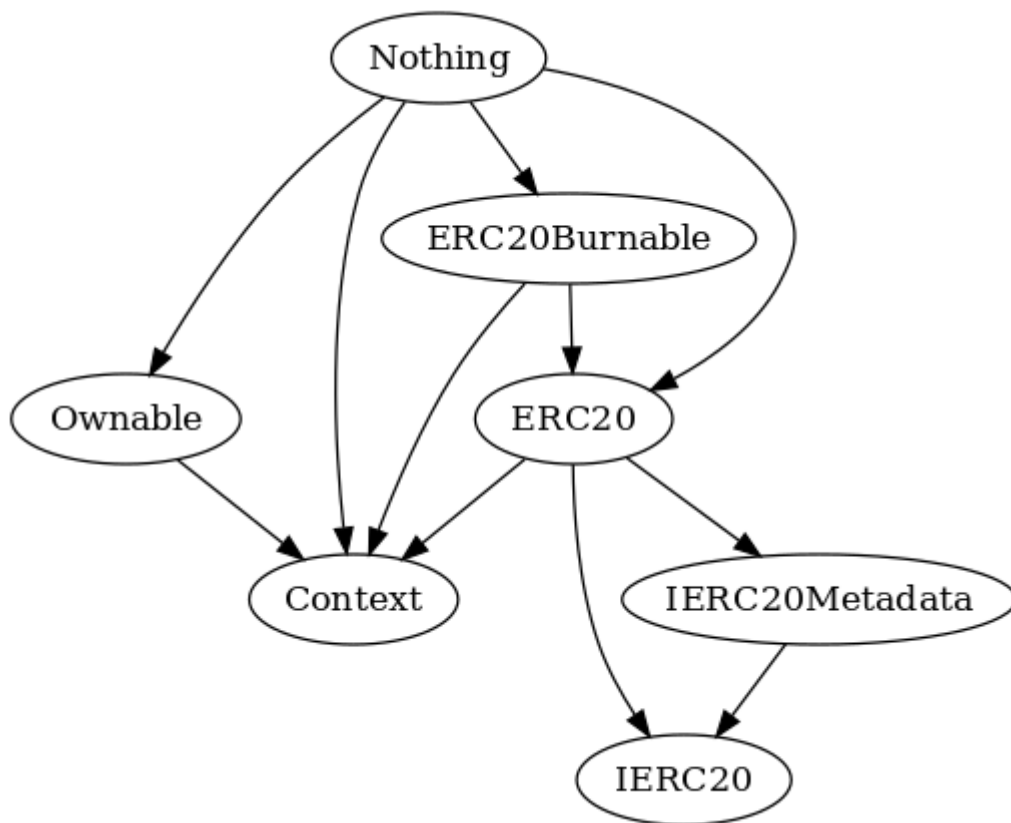
Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

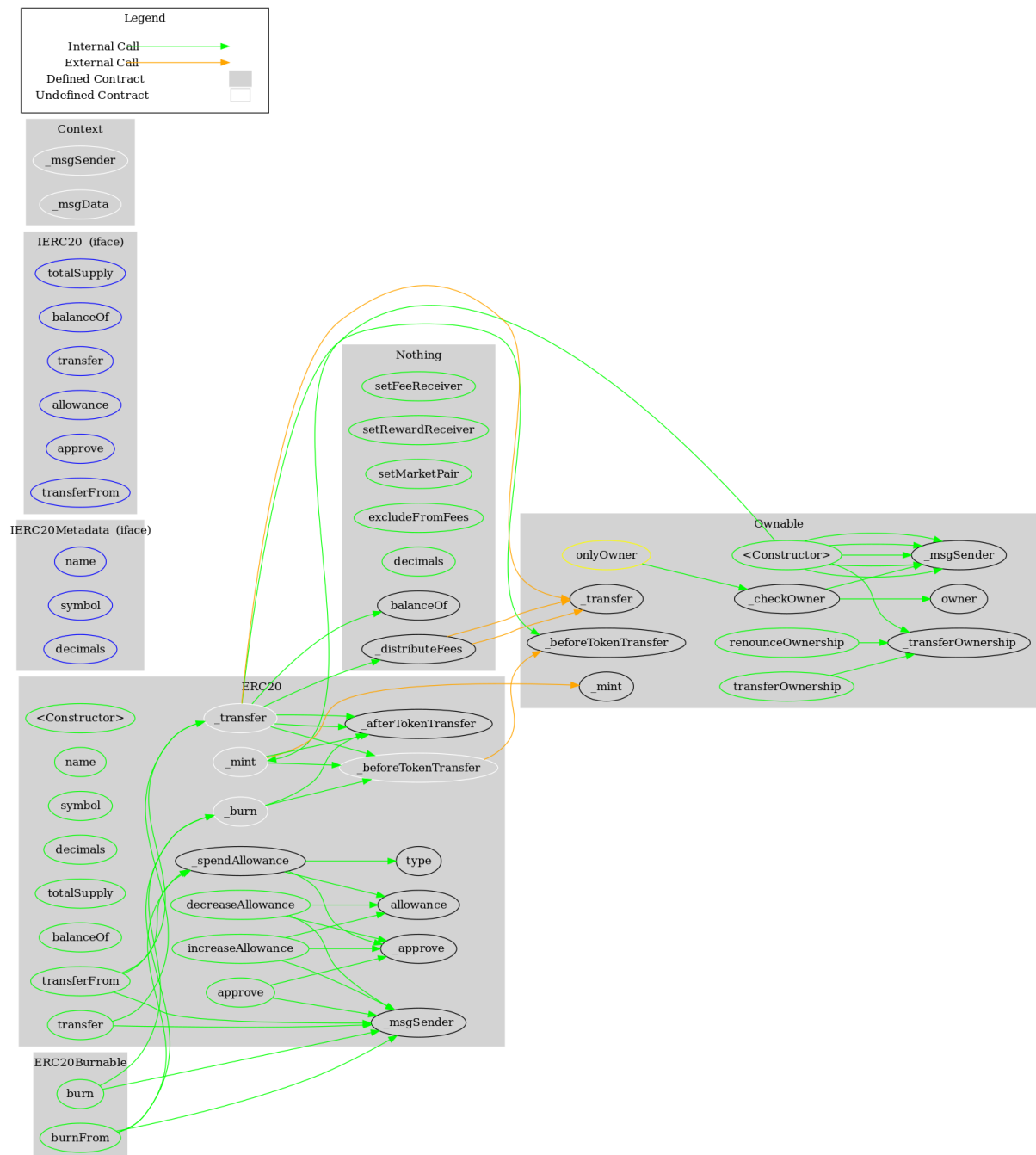
Functions Analysis

| Contract | Type | Bases | | |
|----------|----------------------|--|------------|-----------|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| Nothing | Implementation | Context, ERC20, ERC20Burnable, Ownable | | |
| | | Public | ✓ | ERC20 |
| | setFeeReceiver | Public | ✓ | onlyOwner |
| | setRewardReceiver | Public | ✓ | onlyOwner |
| | setMarketPair | Public | ✓ | onlyOwner |
| | excludeFromFees | Public | ✓ | onlyOwner |
| | decimals | Public | | - |
| | _mint | Internal | ✓ | |
| | _beforeTokenTransfer | Internal | ✓ | |
| | _transfer | Internal | ✓ | |
| | _distributeFees | Internal | ✓ | |

Inheritance Graph



Flow Graph



Summary

ETHnothing contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. ETHnothing is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The Contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 5% fee.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security. Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.



The Cyberscope team

<https://www.cyberscope.io>