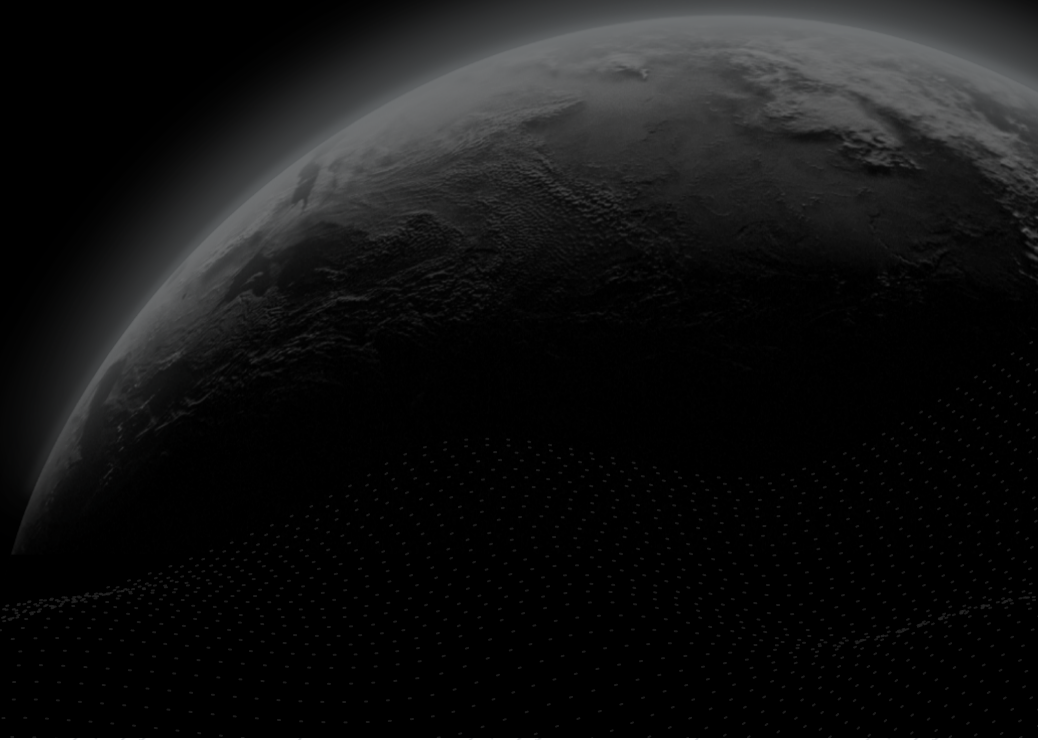
CERTIK

Security Assessment

Draft (Internal Use Only)

Rise of Empire

CertiK Verified on Jul 12th, 2022



CertiK Verified on Jul 12th, 2022

Rise of Empire

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeflationaryToken	BSC	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 07/12/2022	N/A

CODEBASE
0x93bfb96cb8bb094976c53e2b601ca5fbea73df08
...View All

COMMITS
0x93bfb96cb8bb094976c53e2b601ca5fbea73df08
...View All

Vulnerability Summary

9	4	0	1	4	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

1	Critical	1 Resolved	Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
1	Major	1 Acknowledged	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
4	Minor	1 Resolved, 1 Partially Resolved, 2 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
3	Informational	2 Resolved, 1 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | RISE OF EMPIRE

CODEBASE | RISE OF EMPIRE

Repository

0x93bfb96cb8bb094976c53e2b601ca5fbea73df08

Commit

0x93bfb96cb8bb094976c53e2b601ca5fbea73df08

AUDIT SCOPE | RISE OF EMPIRE

0 files audited

ID	File	SHA256 Checksum

APPROACH & METHODS | RISE OF EMPIRE

This report has been prepared for Rise of Empire to discover issues and vulnerabilities in the source code of the Rise of Empire project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

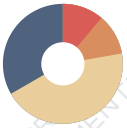
The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

FINDINGS | RISE OF EMPIRE



9	1	1	0	4	3
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Rise of Empire. Through this audit, we have uncovered 9 issues ranging from different severity levels. Utilizing Static Analysis techniques to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
ROE-01	Potential Death Spiral	Logical Issue	Critical	Resolved
ROE-02	**Centralization Risks In ROEMPToken.Sol**	**Centralization / Privilege**	**Major**	**Acknowledged**
ROE-03	Usage Of `transfer()` For Sending Ether	Language Specific	Minor	Acknowledged
ROE-04	Lack Of Reasonable Boundary	Volatile Code	Minor	Resolved
ROE-05	Unchecked ERC-20 `transfer()` / `transferFrom()` Call	Volatile Code	Minor	Acknowledged
ROE-06	Missing Zero Address Validation	Volatile Code	Minor	Partially Resolved
0X9-02	Missing Emit Events	Coding Style	Informational	Acknowledged
ROE-07	Unnecessary Checks	Language Specific	Informational	Resolved
ROE-08	Expression Always True	Logical Issue	Informational	Resolved

ROE-01 | POTENTIAL DEATH SPIRAL

Category	Severity	Location	Status
Logical Issue	Critical	contracts/ROEMPToken.sol: 193~208	Resolved

Description

The _swapAndLiquify() function does three things:

1. Assume rate = 20%.
2. Swap 0.9% of the balance of the liquidity pool(ROEMP/BUSD) for BUSD.
3. Add 0.1% of the balance of the liquidity pool(ROEMP/BUSD) and an amount of BUSD to the pool.
4. Transfer the rest of BUSD to the marketing wallet.

We can see that more ROEMP will be swapped for USD than added to the liquidity pool, so the price of ROEMP will decrease again after each sell. Attackers can write a for loop to sell as little as one ROEMP each time to cause a death spiral.

Recommendation

We recommend setting a limit for triggering the _swapAndLiquify() function. In addition, the business logic needs to be modified so that it is not prone to the economic attack mentioned above.

Alleviation

[Rise of Empire] - All taxes will go into the marketing wallet and no auto swap of USD.

[CertiK] - Since the auto swap feature has been removed the price of the token will not be affected as much, thus resolving this finding. The changes were made in this commit 0x2d2526186598F150f7c94c3dd5A2Ef6e83DF0Ef7

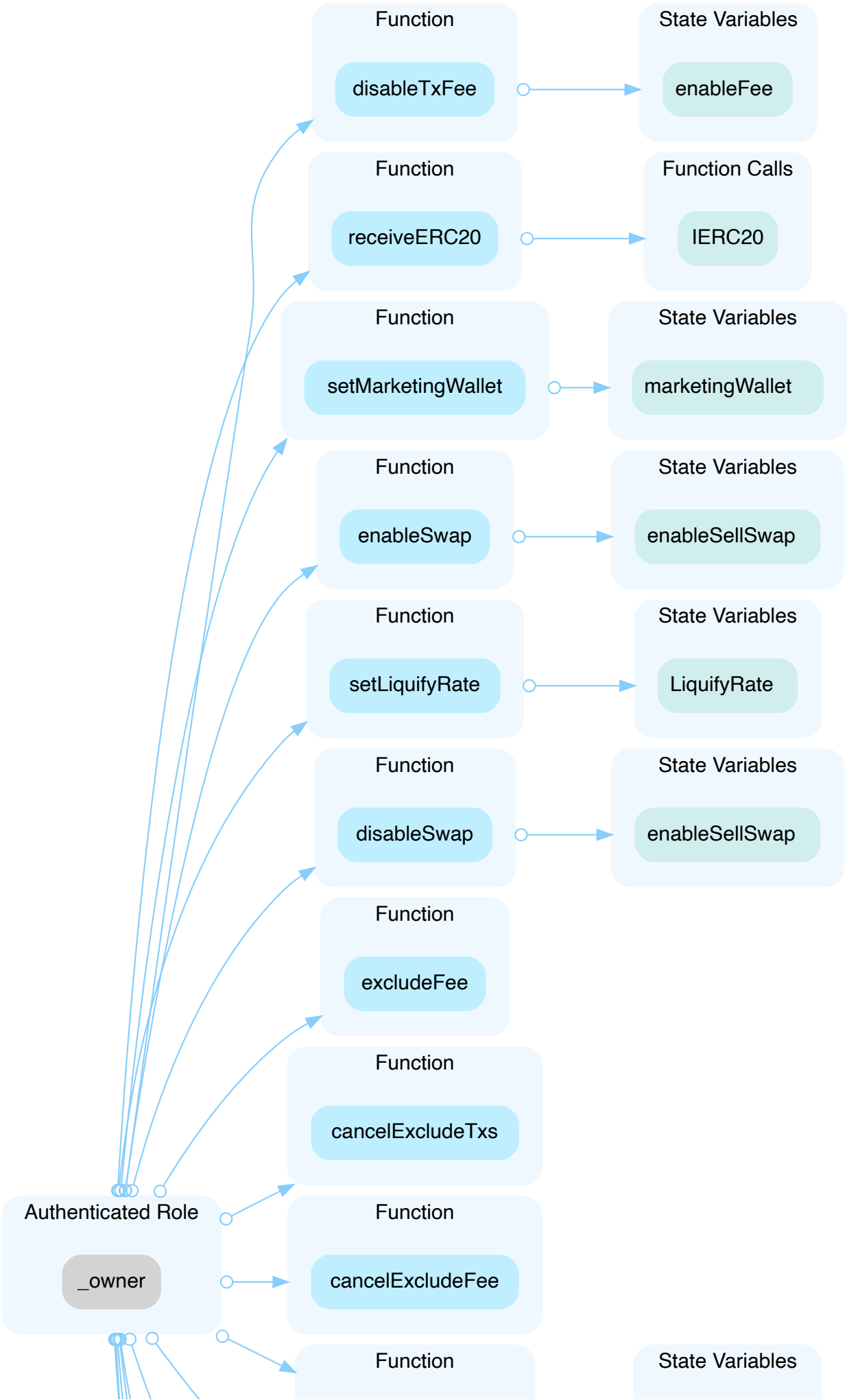
ROE-02 | CENTRALIZATION RISKS IN ROEMPTOKEN.SOL

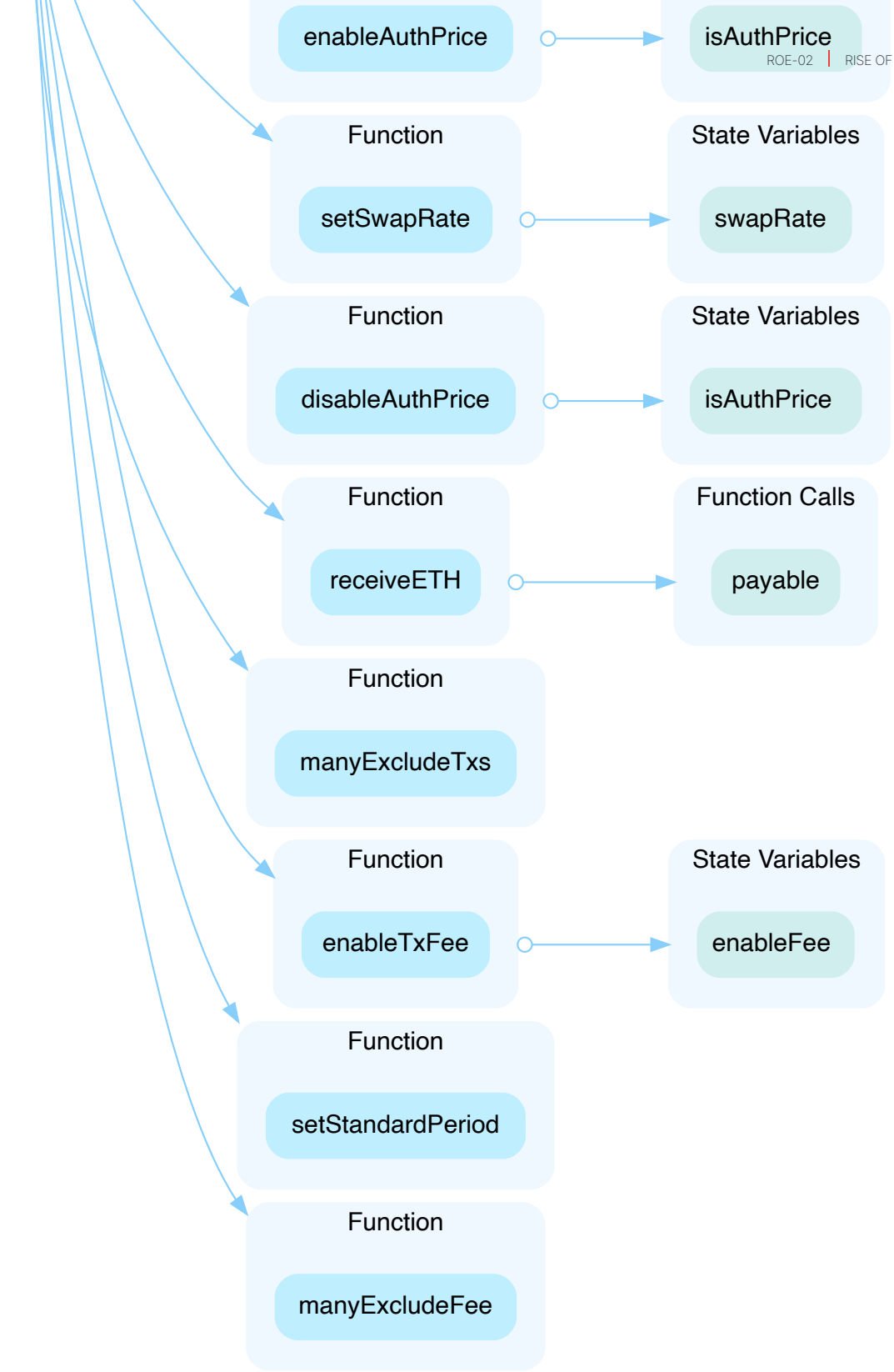
Category	Severity	Location	Status
Centralization / Privilege	**Major**	**contracts/ROEMPToken.sol: 11, 219, 223, 229, 239, 245, 255, 259, 263, 267, 271, 275, 279, 283, 333, 337, 342, 353**	**Acknowledged**

Description

In the contract `ROEMPToken` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and drain all ERC20 tokens and ETH by calling the receiveERC20() and receiveETH() function, respectively.

Also, a compromise to the `_owner` account can implement the attack written in the finding `Lack of Boundary`.


Function
disableTxFee
State Variables
enableFee
Function
receiveERC20
Function Calls
IERC20
Function
setMarketingWallet
State Variables
marketingWallet
Function
enableSwap
State Variables
enableSellSwap
Function
setLiquifyRate
State Variables
LiquifyRate
Function
disableSwap
State Variables
enableSellSwap
Function
excludeFee
Function
cancelExcludeTxs
Authenticated Role
_owner
Function
cancelExcludeFee
Function
State Variables



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

[CertiK] - The Rise of Empire team acknowledged the finding but haven't made any changes to mitigate the finding.

ROE-03 | USAGE OF `transfer()` FOR SENDING ETHER

Category	Severity	Location	Status
Language Specific	Minor	contracts/ROEMPToken.sol: 359~360	Acknowledged

Description

After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` or `.send()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case the linked statements are meant to be able to transfer funds to other contracts instead of EOAs.

Recommendation

We advise that the linked `.transfer()` and `.send()` calls are substituted with the utilization of the `sendValue()` function from the `Address.sol` implementation of OpenZeppelin either by directly importing the library or copying the linked code.

Alleviation

[CertiK] - The client acknowledged the finding but haven't made any changes in regards to this finding.

ROE-04 | LACK OF REASONABLE BOUNDARY

Category	Severity	Location	Status
Volatile Code	Minor	contracts/ROEMPToken.sol: 271~277	Resolved

Description

The variables `swapRate` and `LiquifyRate` do not have reasonable boundaries, so they can be given arbitrary values after deploying. If the `_owner` account is compromised, the attacker can set the `LiquifyRate` to 0. This means this contract will swap all the tokens for BUSD and not provide any liquidity after, leading to a considerable price drop.

Recommendation

We recommend adding reasonable upper and lower boundaries to all the configuration variables.

Alleviation

[Rise of Empire] - We have deleted the auto part for that.

[CertiK] - Since the auto swap feature has been removed these two variables are also deleted, thus resolving this finding. The changes were made in this commit 0x2d2526186598F150f7c94c3dd5A2Ef6e83DF0Ef7

ROE-05 | UNCHECKED ERC-20 `transfer()` / `transferFrom()` CALL

Category	Severity	Location	Status
Volatile Code	Minor	contracts/ROEMPToken.sol: 14, 207, 350	Acknowledged

Description

The return value of the transfer()/transferFrom() call is not checked.

```
14              IERC20(token).transfer(owner(), balance);
```

```
207         usdToken.transfer(marketingWallet, newBalance.sub(liquifyUSD));
```

```
350         token.transfer(to, amount);
```

Recommendation

Since some ERC-20 tokens return no values and others return a `bool` value, they should be handled with care. We advise using the OpenZeppelin's `SafeERC20.sol` implementation to interact with the `transfer()` and `transferFrom()` functions of external ERC-20 tokens. The OpenZeppelin implementation checks for the existence of a return value and reverts if `false` is returned, making it compatible with all ERC-20 token implementations.

Alleviation

[CertiK] - The client acknowledged the finding but haven't made any changes in regards to this finding.

ROE-06 | MISSING ZERO ADDRESS VALIDATION

Category	Severity	Location	Status
Volatile Code	Minor	contracts/ROEMPToken.sol: 280, 359	Partially Resolved

Description

Addresses should be checked before assignment or external call to make sure they are not zero addresses.

```
280         marketingWallet = _marketingWallet;
```

- `_marketingWallet` is not zero-checked before being used.

```
359         to.transfer(amount);
```

- `to` is not zero-checked before being used.

Recommendation

We advise adding a zero-check for the passed-in address value to prevent unexpected errors.

0X9-02 | MISSING EMIT EVENTS

Category	Severity	Location	Status
Coding Style	Informational	@openzeppelin/contracts/access/Ownable.sol: 54, 62; contracts/ROEMPToken.sol: 11, 219, 223, 229, 239, 245, 255, 259, 263, 267, 271, 275, 279, 333, 337, 342, 353	Acknowledged

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Alleviation

[CertiK] - The client acknowledged the finding but haven't made any changes in regards to this finding.

ROE-07 | UNNECESSARY CHECKS

Category	Severity	Location	Status
Language Specific	Informational	contracts/ROEMPToken.sol: 347, 357	Resolved

Description

The check

```
require(token.balanceOf(address(this)) > amount, "Token is running low");`
```

is redundant because it happens in ERC20.sol.

```
    function _transfer(
        address from,
        address to,
        uint256 amount
    ) internal virtual {
        require(from != address(0), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");

        _beforeTokenTransfer(from, to, amount);

        uint256 fromBalance = _balances[from];
        require(fromBalance >= amount, "ERC20: transfer amount exceeds balance"); //
(HERE!!!)
        unchecked {
            _balances[from] = fromBalance - amount;
        }
        _balances[to] += amount;

        emit Transfer(from, to, amount);

        _afterTokenTransfer(from, to, amount);
    }
```

The second check,

```
require(payable(address(this)).balance > amount, "ETH is running low");
```

is also unnecessary because it will be checked by the Ethereum Protocol.

Recommendation

We recommend deleting these blocks to save gas.

Alleviation

[CertiK] - The unnecessary checks were removed from the codebase. This does reduce the gas fees very slightly. The changes can be seen here 0x2d2526186598F150f7c94c3dd5A2Ef6e83DF0Ef7

ROE-08 | EXPRESSION ALWAYS TRUE

Category	Severity	Location	Status
Logical Issue	Informational	contracts/ROEMPToken.sol: 111	Resolved

Description

The linked expression is always True.

Recommendation

We recommend reviewing this logic.

Alleviation

[CertiK] - Since the swap feature has been removed from the codebase this finding is not relevant anymore hence we can set the status of the finding to resolved.

OPTIMIZATIONS | RISE OF EMPIRE

ID	Title	Category	Severity	Status
0X9-01	Unnecessary Use Of SafeMath	Gas Optimization	Optimization	Acknowledged

0X9-01 | UNNECESSARY USE OF SAFEMATH

Category	Severity	Location	Status
Gas Optimization	Optimization	@openzeppelin/contracts/utils/math/SafeMath.sol: 16; contracts/ROEMPToken.sol: 90, 106, 109, 152, 153, 194, 195, 201, 207, 299, 306	Acknowledged

Description

The `SafeMath` library is used unnecessarily. With Solidity compiler versions 0.8.0 or newer, arithmetic operations will automatically revert in case of integer overflow or underflow.

```
16 library SafeMath {
```

- An implementation of `SafeMath` library is found.

```
20     using SafeMath for uint256;
```

- `SafeMath` library is used for `uint256` type in `ROEMPToken` contract.

```
90                 amount = amount.sub(feeAmount);
```

- `SafeMath.sub` is called in `_transfer` function of `ROEMPToken` contract.

Note: Only a sample of 2 `SafeMath` library usage in this contract (out of 20) are shown above.

Recommendation

We advise removing the usage of `SafeMath` library and using the built-in arithmetic operations provided by the Solidity programming language.

Alleviation

[Rise of Empire] - Issue acknowledged. I will fix the issue in the future, which will not be included in this audit engagement.

APPENDIX | RISE OF EMPIRE

Finding Categories

Categories	Description
Centralization / Privilege	Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.
Gas Optimization	Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.
Logical Issue	Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
Language Specific	Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.
Coding Style	Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. 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 CertiK's prior written consent in each instance.

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 considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report 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 in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK'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 CertiK is subject to dependencies and 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.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE,

OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.