



## SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



NEXTEP  
\$NEXTEP



22/01/2022



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7-8 OWNER PRIVILEGES
- 9 CONCLUSION AND ANALYSIS
- 10 TOKEN DETAILS
- 11 NEXTEP TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 12 TECHNICAL DISCLAIMER



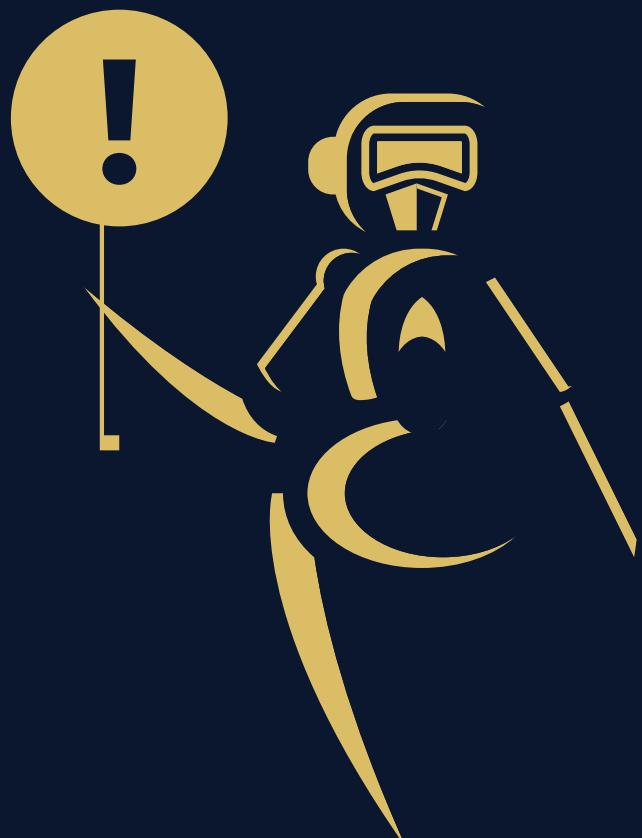
# DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy ( RUG or Honeypot etc )



# INTRODUCTION

**FreshCoins (Consultant) was contracted by  
NEXTEP (Customer) to conduct a Smart Contract Code Review  
and Security Analysis.**

**0x477e701Fb4A3B98C0eE38AE75d8C6349BAF46e8F**

**Network: Binance Smart Chain (BSC)**

**This report presents the findings of the security assessment of  
Customer's smart contract and its code review conducted on 22/01/2022**



# AUDIT OVERVIEW



**Security Score**



**Static Scan**  
Automatic scanning for common vulnerabilities



**ERC Scan**  
Automatic checks for ERC's conformance



**High**



**Medium**



**Low**



**Optimizations**



**Informational**



| No. | Issue description              | Checking Status |
|-----|--------------------------------|-----------------|
| 1   | Compiler Errors / Warnings     | Passed          |
| 2   | Reentrancy and Cross-function  | Passed          |
| 3   | Front running                  | Passed          |
| 4   | Timestamp dependence           | Passed          |
| 5   | Integer Overflow and Underflow | Passed          |
| 6   | Reverted DoS                   | Passed          |
| 7   | DoS with block gas limit       | Low             |
| 8   | Methods execution permissions  | Passed          |
| 9   | Exchange rate impact           | Passed          |
| 10  | Malicious Event                | Passed          |
| 11  | Scoping and Declarations       | Passed          |
| 12  | Uninitialized storage pointers | Passed          |
| 13  | Design Logic                   | Passed          |
| 14  | Safe Zeppelin module           | Passed          |

# OWNER PRIVILEGES

**Contract owner can't mint tokens after initial contract deploy.**

**Owner must be a contract with transparent rules for using openTrade, stopTrade functions.**

**Contract owner can exclude/include wallet from fee**

```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

**Contract owner can change max tx amount and exclude wallet for tx amount limitations**

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    _maxTxAmount = maxTxAmount;
}
.

.

.

function setExcludeFromMaxTxLimit(address account, bool _enabled) public onlyOwner {
    _isExcludedFromMaxTxLimit[account] = _enabled;
}
```

**Contract owner is able to blacklist wallet(s)**

```
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
}
```

**Contract owner can change sell fees**

```
function setAllSaleFeesPercents(uint256 liquidityFee, uint256 marketingFee, uint256 developerFee, uint256 burnFee) external onlyOwner {
    _saleLiquidityFee = liquidityFee;
    _saleMarketingFee = marketingFee;
    _saleDeveloperFee = developerFee;
    _saleBurnFee = burnFee;
}
```

## Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {  
    emit OwnershipTransferred(_owner, address(0));  
    _owner = address(0);  
}
```

## Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {  
    require(newOwner != address(0), "Ownable: new owner is the zero address");  
    emit OwnershipTransferred(_owner, newOwner);  
    _owner = newOwner;  
}
```

## Contract owner can change the fees

```
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {  
    _liquidityFee = liquidityFee;  
    _previousLiquidityFee = liquidityFee;  
}  
  
function setMarketingFeePercent(uint256 marketingFee) external onlyOwner() {  
    _marketingFee = marketingFee;  
    _previousMarketingFee = marketingFee;  
}  
  
function setDeveloperFeePercent(uint256 developerFee) external onlyOwner() {  
    _developerFee = developerFee;  
    _previousDeveloperFee = developerFee;  
}  
  
function setBurnFeePercent(uint256 burnFee) external onlyOwner() {  
    _burnFee = burnFee;  
    _previousBurnFee = burnFee;  
}
```

## Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {  
    swapAndLiquifyEnabled = _enabled;  
    emit SwapAndLiquifyEnabledUpdated(_enabled);  
}
```

### Recomandation:

**The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. The risk can be prevented by temporarily locking the contract or renouncing ownership.**

# CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.



Found 1 LOW issue during the first review.

# TOKEN DETAILS

## Details

|            |              |
|------------|--------------|
| Buy fees:  | 20%          |
| Sell fees: | 20%          |
| Max TX:    | 100000000000 |
| Max Sell:  | N/A          |

## Honeypot Risk

|                  |              |
|------------------|--------------|
| Ownership:       | Owned        |
| Blacklist:       | Detected     |
| Modify Max TX:   | Detected     |
| Modify Max Sell: | Not detected |
| Disable Trading: | Not detected |

## Rug Pull Risk

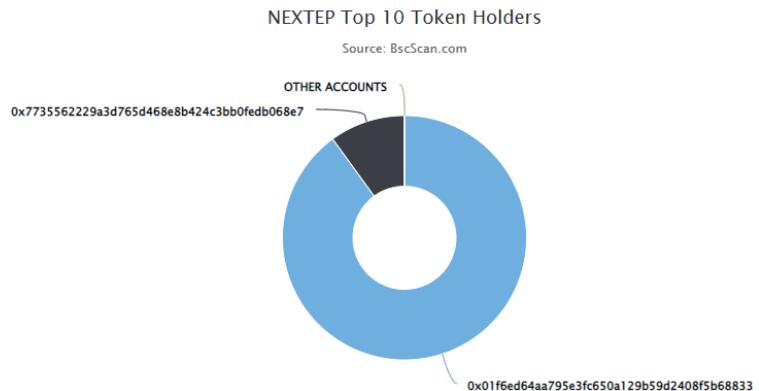
|            |       |
|------------|-------|
| Liquidity: | N/A   |
| Holders:   | Clean |



# NEXTEP TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of NEXTEP

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 2



(A total of 100,000,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

| Rank | Address                                    | Quantity (Token) | Percentage |
|------|--|------------------|------------|
| 1    | 0x01f6ed64aa795e3fc650a129b59d2408f5b68833 | 90,000,000,000   | 90.0000%   |
| 2    | 0x7735562229a3d765d468e8b424c3bb0fdb068e7  | 10,000,000,000   | 10.0000%   |

# TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

