



# **BCA Blockchain Contract Audit**

區塊鏈合約檢測服務

Project: MasterChef

Report date: Aug 17th, 2023

- ✓ Re-entrancy
- ✓ Overflow/underflow
- ✓ Use of block.timestamp
- ✓ Use of tx.origin
- ✓ Use of selfdestruct
- ✓ Storage conflict
- ✓ Force receive token
- ✓ Using inline assembly
- ✓ Access vulnerability
- ✓ Return value of low level call
- ✓ Return value of transfer
- ✓ Enable Trade

**Audit Status: PASS**

<b>Vulnerabilities list</b>	p.02
-----------------------------	------

<b>Summary</b>	p.04
----------------	------

<b>Vulnerabilities Check</b>	p.06
------------------------------	------

<b>Conclusion</b>	p.12
-------------------	------

<b>Disclaimer</b>	p.13
-------------------	------

Project name	MasterChef
Network	opBNB
Language	Solidity
Delivery Date	2023/8
Contract Address	0x39a786421889EB581bd105508a0D2Dc03523B903

This report only covers a check of common vulnerabilities mentioned in the above list. Many critical vulnerabilities or malicious backdoors may not be detected. To undertake a more comprehensive inspection, we recommend upgrading to our full-service version.

	Token Information (Not Applicable)
Fee	Not Applicable
Fee Privilege	Not Applicable
Ownership	Not Applicable
Max Tx Amount	Not Applicable
Blacklist	Not Applicable
Decimals	Not Applicable
Max Supply	Not Applicable
Mint/Burn	Not Applicable

# Re-entrancy

If a contract has this vulnerability, when it calls an external contract, and does not update its status before sending funds, an attacker could continually call the withdraw function to transfer funds until all funds in the contract are depleted.

**PASS**

# Overflow/underflow

When performing calculations on numbers, if the result exceeds or falls below the range of the type, an Overflow or Underflow vulnerability can occur.

**PASS**

# Dependence on `block.timestamp`

Generating random numbers using global variables like `timestamp` can be predicted by attackers.

**NONE**

# Use of `tx.origin`

When a contract uses `tx.origin` to verify user identity, malicious actors can exploit this vulnerability, masquerading as an address that can pass verification.

**NONE**

# Use of selfdestruct

When a contract improperly uses the selfdestruct function, it can result in the contract being destroyed and its balance transferred to an address controlled by the attacker.

**NONE**

# Storage conflict

If different variables share the same storage slot, it can lead to variables being maliciously altered by attacker.

**PASS**



# Force receive token

If the balance of the contract is used as a check condition, the contract may become invalid if an attacker forces a transfer.

**PASS**

# Using inline assembly

The use of assembly is error-prone and should be avoided.

**NONE**

# Access vulnerability

Vulnerabilities in permissions may allow malicious actors to bypass identity checks for accessing functions, or to change the owner of the permissions.

**PASS**

## Return value of low level call

This vulnerability refers to an issue where, during the execution of `call()`, a return value is typically given to indicate whether the function was successful or not. If this return value is not properly used, unexpected errors may occur.

**NONE**

# Return value of transfer

This vulnerability refers to an issue where, during the execution of `transfer()`, a return value is typically given to indicate whether the transfer was successful or not. If this return value is not properly used, unexpected errors may occur.

**NONE**

# Enable Trade

If the contract includes the "Enable Trade" feature, the project party has the right to disable users' token trading privileges. The users' assets will be at risk.

**NONE**

# Conclusion

This is an implementation of the Defi LP token staking pool that has been audited and found to have no vulnerabilities from the list.

The audit scope does not encompass the contract for reward token. Investors are advised to exercise caution and beware of the risks involved.

# Disclaimer

Before you use this website to fill in basic information, upload information and apply to this service, you have to read this Terms of Service on the website thoroughly to protect your right.

We only audit common hacking issues in the above smart contracts, and do not guarantee the business model of this project. Investment involves risks, please consider carefully before purchasing.

