區塊鏈合約檢測服務

Project: Oscars20 Token

Report date: Jul 9th, 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: FAIL

Project name	Oscars20 Token
Network	BSC
Language	Solidity
Delivery Date	2023/7
Contract Address	0xF12532F433d637EaB1d3c02D115BB672C1b414d9

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
Fee	Yes
Fee Privilege	Yes
Ownership	Yes
Max Tx Amount	None
Blacklist	None
Decimals	9
Max Supply	One Billion
Mint/Burn	None

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.

FAIL **Oscars20.sol: 569-578**

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

Dependance on block.timestamp

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

PASS

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.

YES

Conclusion

This is an implementation of the ERC20 token standard. This token includes the enableTrading function, granting the contract owner the authority to halt token trading by users. If the _owner address is a malicious contract, the function claimStuckTokens() may trigger reentrancy attack. Investors are advised to exercise caution and be aware 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.

Blockchain
Contract Audit