Audit Report for **SquidTokenV2**
Date: 08 April 2024
Audit Result: **Passed with medium risk.**

Token Address: 0xFAfb7581a65A1f554616Bf780fC8a8aCd2Ab8c9b

Name: Squid Game V2

Symbol: SQUID

Decimals: 18

Network: Base Scan

Token Type: ERC-20

Owner: 0xff00d2D6210a537B517138389C29c8A6bb56DaD7

Deployer: 0xff00d2D6210a537B517138389C29c8A6bb56DaD7

Token Supply: 71,430,240

Checksum: A2032c616934aeb47e6039f76b20d261

Testnet:
https://testnet.bscscan.com/address/0x98dE47C1a577558356a702a5267633cC133741dD#code

Token Overview:

Buy Fee: 0%

Sell Fee: 0%

Transfer Fee: 0%

Fee Privilege: Owner

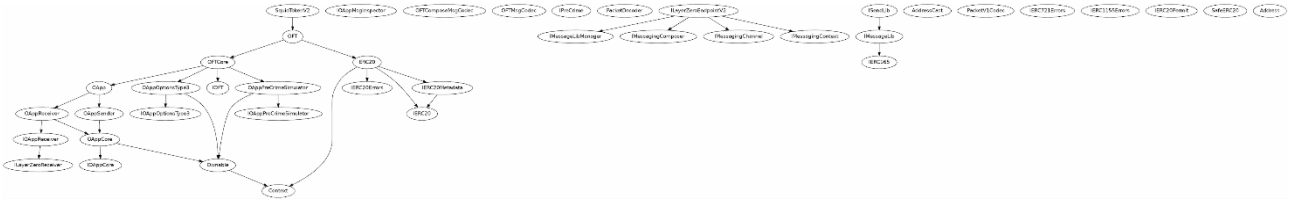
Ownership: Owned

Minting: None

Max Tx: No

Blacklist: No

Inheritance Tree



Static Analysis

A static analysis of the code was performed using Slither. No issues were found.

Functional Tests
Router (PCS V2):

1- Approve (passed):
https://testnet.bscscan.com/tx/0xfea5aaa7c9cdbdc3b4f88cbcfc9acfd0ad4b6c3c142f7548372d95919451339a

2- Set Delegate (passed):
https://testnet.bscscan.com/tx/0x71cb446be9eabba57b9d3a9c4c09cf46a2bdc83eb07e7b8ca1934a129e454cd7

3- Set Pre-Crime (passed):
https://testnet.bscscan.com/tx/0xa7577bfa4f3ab070a1efdf6955e4106781a96d03ae6cfefff238f67e1d3f33f0

4- Set Msg Inspector (passed):
https://testnet.bscscan.com/tx/0x8868451110c8428311bd42f016fb45a9beefb422d49f5935c195acac0eee6268

Ownership Privileges:

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can set delegate.
- The owner can set peer.
- The owner can set pre-crime.
- The owner can set Msg Inspector.

Findings:
Critical: 0
High: 0
Medium: 1
Low: 1
Informational & Optimizations: 1

Centralization – Missing Require Check.

Severity: Medium
Function: Set Delegate/Msg Inspector
Status: Open
Overview:
The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMsgInspector(address _msgInspector) public virtual onlyOwner {
        msgInspector = _msgInspector;
        emit MsgInspectorSet(_msgInspector);
    }
function setDelegate(address _delegate) external;
}
```

Suggestion:
It is recommended that the address should not be able to be set as a contract address.

Centralization – Missing Events

Severity: Low
Subject: Missing Events
Status: Open
Overview:
They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setDelegate(address _delegate) external;
}
```

Suggestion:
Emit an event for critical changes.

Optimization

Severity: Optimization
Subject: Remove unused code.
Status: Open
Overview:
Unused variables are allowed in Solidity, and they do not pose a direct security issue. It is the best practice though to avoid them.

```
function sendValue(address payable recipient, uint256 amount) internal {
        if (address(this).balance < amount) {
            revert AddressInsufficientBalance(address(this));
        }

        (bool success, ) = recipient.call{value: amount}("");
        if (!success) {
            revert FailedInnerCall();
        }
    }
function functionCall(address target, bytes memory data) internal returns (bytes
memory) {
        return functionCallWithValue(target, data, 0);
    }
function functionStaticCall(address target, bytes memory data) internal view re-
turns (bytes memory) {
        (bool success, bytes memory returndata) = target.staticcall(data);
        return verifyCallResultFromTarget(target, success, returndata);
    }
function functionDelegateCall(address target, bytes memory data) internal returns
(bytes memory) {
        (bool success, bytes memory returndata) = target.delegatecall(data);
        return verifyCallResultFromTarget(target, success, returndata);
    }
library AddressCast {
    error AddressCast_InvalidSizeForAddress();
    error AddressCast_InvalidAddress();

function toBytes32(bytes calldata _addressBytes) internal pure returns (bytes32 re-
sult) {
        if (_addressBytes.length > 32) revert AddressCast_InvalidAddress();
        result = bytes32(_addressBytes);
        unchecked {
            uint256 offset = 32 - _addressBytes.length;
            result = result >> (offset * 8);
        }
    }

function toBytes32(address _address) internal pure returns (bytes32 result) {
        result = bytes32(uint256(uint160(_address)));
    }
```

```
function toBytes(bytes32 _addressBytes32, uint256 _size) internal pure returns
(bytes memory result) {
        if (_size == 0 || _size > 32) revert AddressCast_InvalidSizeForAddress();
        result = new bytes(_size);
        unchecked {
            uint256 offset = 256 - _size * 8;
            assembly {
                mstore(add(result, 32), shl(offset, _addressBytes32))
            }
        }
    }

function toAddress(bytes32 _addressBytes32) internal pure returns (address result)
{
        result = address(uint160(uint256(_addressBytes32)));
    }

function toAddress(bytes calldata _addressBytes) internal pure returns (address re-
sult) {
        if (_addressBytes.length != 20) revert AddressCast_InvalidAddress();
        result = address(bytes20(_addressBytes));
    }
}
abstract contract Context {
    function _msgSender() internal view virtual returns (address) {
        return msg.sender;
    }

    function _msgData() internal view virtual returns (bytes calldata) {
        return msg.data;
    }

    function _contextSuffixLength() internal view virtual returns (uint256) {
        return 0;
    }
}
interface IERC165 {
    function supportsInterface(bytes4 interfaceId) external view returns (bool);
}
```

Suggestion:

To reduce high gas fees. It is suggested to remove unused code from the contract.