



# Coinsult

## Advanced Manual Smart Contract Audit



**Project:** ShinobiVerse

**Website:** [www.shinobiverse.io](http://www.shinobiverse.io)

**Low-risk**

5 low-risk code  
issues found

**Medium-risk**

0 medium-risk code  
issues found

**High-risk**

0 high-risk code  
issues found

**Contract address**

0x24827DBcaC8dDEbC1DEE746A1e432eeA9A6c2a24

Disclaimer: Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

# Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

# Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xc4f07cc01593687c4d645de74942b2309d686886	45,000,000,000	100.0000%

# Source code

Coinsult was commissioned by ShinobiVerse to perform an audit based on the following smart contract:

<https://cronoscan.com/address/0x24827DBcaC8dDEbC1DEE746A1e432eeA9A6c2a24#code>

# Manual Code Review

## ● Low-risk

5 low-risk code issues found.

Could be fixed, will not bring problems.

- Contract contains Reentrancy vulnerabilities:

`_transfer(address,address,uint256)`

Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback).

More information: [Slither](#)

```
function _transfer(
    address from,
    address to,
    uint256 amount
) private {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");
    require(amount > 0, "Transfer amount must be greater than zero");

    if (from != owner() && to != owner() && ! _isExcludedFromFee[to] && ! _isExcludedFromFee[from]) {

        if (from == uniswapV2Pair && to != address(uniswapV2Router) && ! _isExcludedFromFee[to]) {
            require(amount <= _bMaxTxAmount, "Transfer amount exceeds max buy amount.");
        }

        if (to == uniswapV2Pair && ! _isExcludedFromFee[from]){
            require(amount <= _sMaxTxAmount, "Transfer amount exceeds the max sell amount.");
        }
    }

    uint256 contractTokenBalance = balanceOf(address(this));
```





- Block.timestamp can be manipulated by miners.  
Avoid relying on block.timestamp.

More information:

<https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp>

```
uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(  
    tokenAmount,  
    0, // Accept any amount of CRO  
    path,  
    address(this), // The contract  
    block.timestamp  
);
```

- Missing zero address validation.  
Check that the new address is not zero.

```
function setMarketingAddress(address _marketingAddress) external  
onlyOwner {  
    marketingAddress = payable(_marketingAddress);  
    _isExcludedFromFee[marketingAddress] = true;  
}  
  
function setDevelopmentAddress(address _developmentAddress)  
external onlyOwner {  
    developmentAddress = payable(_developmentAddress);  
    _isExcludedFromFee[developmentAddress] = true;  
}  
  
function setVaultAddress(address _vaultAddress) external onlyOwner  
{  
    vaultRewardAddress = payable(_vaultAddress);  
    _isExcludedFromFee[vaultRewardAddress] = true;  
}
```

- Literals with many digits are difficult to read and review.  
Use: Ether suffix, Time suffix, or The scientific notation

```
uint256 public _bMaxTxAmount = 45000000000 * 10**9;  
uint256 public _sMaxTxAmount = 45000000000 * 10**9;  
uint256 public minimumTokensBeforeSwap = 100000 * 10**9;
```

- Prefer not to use else statement but only else if

```
function _tokenTransfer(address sender, address recipient, uint256
amount) private {
    if (_isExcluded[sender] && !_isExcluded[recipient]) {
        _transferFromExcluded(sender, recipient, amount);
    } else if (!_isExcluded[sender] && _isExcluded[recipient]) {
        _transferToExcluded(sender, recipient, amount);
    } else if (_isExcluded[sender] && _isExcluded[recipient]) {
        _transferBothExcluded(sender, recipient, amount);
    } else {
        _transferStandard(sender, recipient, amount);
    }
}
```

## ● Medium-risk

0 medium-risk code issues found.

Should be fixed, could bring problems.

## ● High-risk

0 high-risk code issues found

Must be fixed, and will bring problems.

## Extra notes by the team

- Owner can exclude addresses from fees.
- Owner can set max transaction amount without constraints
- The ownership of the contract isn't renounced.
- Owner can change the router address
- Fees can be set up to 100% for both buy and sell fees.
- Dev notes can be deleted upon deployment



# Contract Snapshot

```
contract ShinobiVerse is Context, IERC20, Ownable {
    using SafeMath for uint256;
    using Address for address;
    // Multisig Protocol Wallets
    address payable public marketingAddress =
payable(0xB414a204A1478FF6B28229e24432071205Ed5770);
    address payable public vaultRewardAddress =
payable(0x6eFD2C32C74aFd3f199ADB6C19EaC0Afe2cDBb1A);
    address payable public developmentAddress =
payable(0x5d70a5713a36e83690D8367826f6ee6a0C160ea3);

    address payable public usdcAddress =
payable(0xc21223249CA28397B4B6541dfFaEcC539BfF0c59);

    address payable public liquidityWallet = payable(address(this));
    address public deadAddress =
0x0000000000000000000000000000000000000000dEaD;

    mapping (address => uint256) private _rOwned;
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private
_allowances;

    mapping (address => bool) private _isExcludedFromFee;
    mapping (address => bool) private _isExcluded;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private constant _tTotal = 45000000000 * 10**9;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private constant _name = "ShinobiVerse";
    string private constant _symbol = "SHINO";
    uint8 private constant _decimals = 9;
```

# Website Review



Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.

- Mobile Friendly
- Contains no jQuery errors
- SSL Secured
- No major spelling errors

Loading speed: 72%

# Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

- Locked Liquidity - No liquidity yet
- Large unlocked wallets - Tokens not yet distributed
- Doxxed Team (KYC)

# Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

- Ability to sell
- Owner is not able to pause the contract
- Router can be changed

**Note:** Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.