Coinsult

Advanced Manual Smart Contract Audit

Project: Strike Crypto BSC

Website: https://strikecryptobsc.com/eng-us/

Low-Risk	Medium-Risk	High-Risk
5 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract Address

0x47B196a23B230f28e0495EF1693b4efdFB70E451

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	0x455f3ee7fb28addd87eeff757fbc3f7b8cf317fd	100,000,000	100.0000%

Source Code

Coinsult was comissioned by Strike Crypto BSC to perform an audit based on the following smart contract:

https://bscscan.com/address/0x47B196a23B230f28e0495EF1693b4efdFB70E451#code

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

5 low-risk code
issues found

Medium-Risk

0 medium-risk code
issues found

High-Risk

0 high-risk code
issues found

The detailed report continues on the next page...

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Contract contains Reentrancy vulnerabilities

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

```
State variables written after the call(s):
- administrationWallet = 0x2Bf7105Cd8aCf6F5b64548DE2A2eA8acF0983b86 (Strike_Crypto_Metaverse.sol#88)
- developingWallet = 0x2Bf7105Cd8aCf6F5b64548DE2A2eA8acF0983b86 (Strike_Crypto_Metaverse.sol#89)
- exemptFee[address(this)] = true (Strike_Crypto_Metaverse.sol#78)
- exemptFee[DEAD] = true (Strike_Crypto_Metaverse.sol#82)
- exemptFee[ownerWallet] = true (Strike_Crypto_Metaverse.sol#86)
- exemptFee[administrationWallet] = true (Strike_Crypto_Metaverse.sol#91)
- exemptFee[developingWallet] = true (Strike_Crypto_Metaverse.sol#93)
- liquidityPool = WBNB_TOKEN_PAIR (Strike_Crypto_Metaverse.sol#74)
- swapHelper = new SwapHelper() (Strike_Crypto_Metaverse.sol#95)
```

Recommendation

Apply the check-effects-interactions pattern.

Exploit scenario

```
function withdrawBalance(){
    // send userBalance[msg.sender] Ether to msg.sender
    // if mgs.sender is a contract, it will call its fallback function
    if( ! (msg.sender.call.value(userBalance[msg.sender])() ) ){
        throw;
    }
    userBalance[msg.sender] = 0;
}
```

Bob uses the re-entrancy bug to call withdrawBalance two times, and withdraw more than its initial deposit to the contract.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

No zero address validation for some functions

Detect missing zero address validation.

```
function setAdministrationWallet(address account) public isAuthorized(0) { administrationWallet = ac
```

Recommendation

Check that the new address is not zero.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

Bob calls `updateOwner` without specifying the `newOwner`, soBob loses ownership of the contract.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Functions that send Ether to arbitrary destinations

Unprotected call to a function sending Ether to an arbitrary address.

```
function safeWithdraw() external isAuthorized(0) {
  uint256 contractBalance = address(this).balance;
  payable(_msgSender()).transfer(contractBalance);
}
```

Recommendation

Ensure that an arbitrary user cannot withdraw unauthorized funds.

Exploit scenario

```
contract ArbitrarySend{
    address destination;
    function setDestination(){
        destination = msg.sender;
    }

    function withdraw() public{
        destination.transfer(this.balance);
    }
}
```

Bob calls `setDestination` and `withdraw`. As a result he withdraws the contract's balance.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
// Excempt Controllers
function setExemptFee(address account, bool operation) public isAuthorized(2) { exemptFee[account] =
function setExemptFeeReceiver(address account, bool operation) public isAuthorized(2) { exemptFeeRec

// Special Wallets
function setAdministrationWallet(address account) public isAuthorized(0) { administrationWallet = ac
function setDevelopingWallet(address account) public isAuthorized(0) { developingWallet = account; }
```

Recommendation

Emit an event for critical parameter changes.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

`updateOwner()` has no event, so it is difficult to track off-chain changes in the buy price.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Conformance to Solidity naming conventions

Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
Variable ERC20._balances (ERC20.sol#35) is not in mixedCase
Contract Strike_Crypto_Metaverse (Strike_Crypto_Metaverse.sol#11-322) is not in CapWords
Function Strike_Crypto_Metaverse.setWBNB_TOKEN_PAIR(address) (Strike_Crypto_Metaverse.sol#316) is no
Function Strike_Crypto_Metaverse.setWBNB_BUSD_Pair(address) (Strike_Crypto_Metaverse.sol#317) is not
Function Strike_Crypto_Metaverse.getWBNB_TOKEN_PAIR() (Strike_Crypto_Metaverse.sol#318) is not in mi
Function Strike_Crypto_Metaverse.getWBNB_BUSD_Pair() (Strike_Crypto_Metaverse.sol#319) is not in mix
Constant Strike_Crypto_Metaverse._name (Strike_Crypto_Metaverse.sol#17) is not in UPPER_CASE_WITH_UN
Constant Strike_Crypto_Metaverse._symbol (Strike_Crypto_Metaverse.sol#18) is not in UPPER_CASE_WITH_
Constant Strike_Crypto_Metaverse.decimal (Strike_Crypto_Metaverse.sol#21) is not in UPPER_CASE_WITH_
Constant Strike_Crypto_Metaverse.decimalBUSD (Strike_Crypto_Metaverse.sol#22) is not in UPPER_CASE_W
Constant Strike_Crypto_Metaverse.maxSupply (Strike_Crypto_Metaverse.sol#23) is not in UPPER_CASE_WITH
Variable Strike_Crypto_Metaverse._maxTxAmount (Strike_Crypto_Metaverse.sol#25) is not in mixedCase
Variable Strike_Crypto_Metaverse._maxAccountAmount (Strike_Crypto_Metaverse.sol#26) is not in mixedC
Variable Strike_Crypto_Metaverse.WBNB_BUSD_PAIR (Strike_Crypto_Metaverse.sol#48) is not in mixedCase
Variable Strike_Crypto_Metaverse.WBNB_TOKEN_PAIR (Strike_Crypto_Metaverse.sol#49) is not in mixedCas
```

Recommendation

Follow the Solidity naming convention.

Rule exceptions

- Allow constant variable name/symbol/decimals to be lowercase (`ERC20`).
- Allow _ at the beginning of the `mixed_case` match for private variables and unused parameters.

Owner privileges

- 🟢 Owner cannot set fees higher than 25%
- 🟢 Owner cannot pause trading
- 🟢 Owner cannot change max transaction amount
- 🟡 Owner can exclude from fees
- ⚠️ Owner is able to burn tokens
- ⚠️ Owner is able to revoke certain permissions
- ⚠️ Owner is able to give addresses certain permissions

Extra notes by the team

No notes

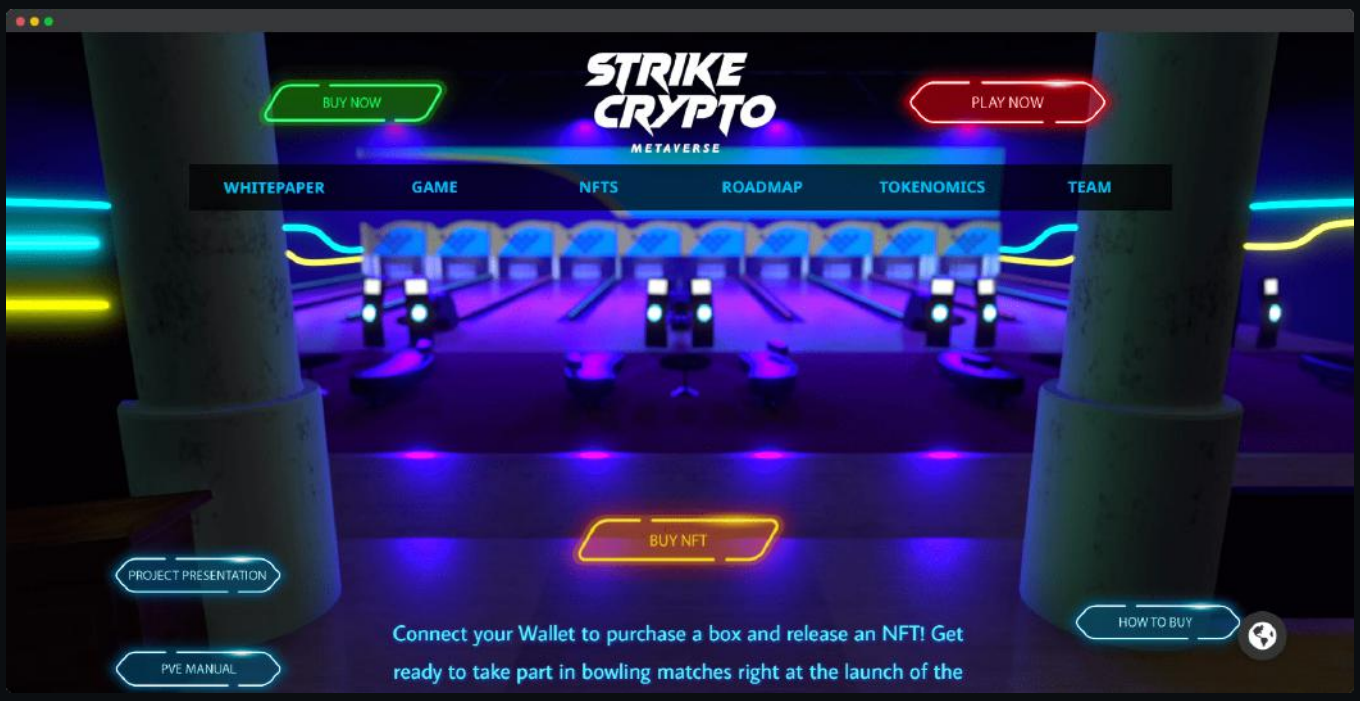
Contract Snapshot

```
contract Strike_Crypto_Metaverse is Authorized, ERC20 {
address constant DEAD = 0x000000000000000000000000000000000000dEaD;
address constant ZERO = 0x0000000000000000000000000000000000000000;
address constant BUSD = 0xe9e7CEA3DedcA5984780Bafc599bD69ADd087D56;
address constant WBNB = 0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c;

string constant _name = "Strike Crypto Metaverse";
string constant _symbol = "STR";
```

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.



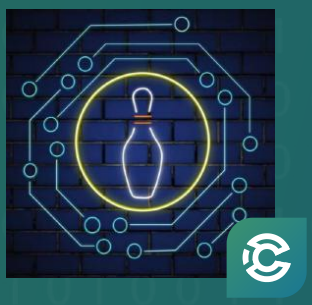
- Mobile Friendly
- Does not contain jQuery errors
- SSL Secured
- No major spelling errors

Project Overview

Not KYC verified by Coinsult

Strike Crypto BSC

Audited by Coinsult.net



Date: 8 July 2022

Advanced Manual Smart Contract Audit