Coinsult

Advanced Manual Smart Contract Audit

Project: Eleia

Website: https://eleia.game/

🟢 Low-Risk	🟡 Medium-Risk	🔴 High-Risk
3 low-risk code issues found	1 medium-risk code issues found	0 high-risk code issues found

Contract Address

0x97F10F99461048C6689e1e04c8b61Ae8332eac49

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	0x1aed86440b5065d523302d9fe1dd24b9044482e3	23,800,000	59.3812%
2	0xc1a2a162931f549881c9ac1e9b24e193e4b86821	7,905,900	19.7253%
3	0x6863e714f3773401004e2c901985f85e80e6c575	3,694,102.6405925	9.2168%
4	0xf4bb01e79505f697c480c8e3710beb07a4b7131d	81,883.364355	0.2043%
5	0xf2f7c54f58a7f6d233df960d307ba57545541325	81,883.364355	0.2043%

Source Code

Coinsult was comissioned by Eleia to perform an audit based on the following smart contract:

https://bscscan.com/address/0x97F10F99461048C6689e1e04c8b61Ae8332eac49#code

Manual Code Review

In this audit report we will highlight all these issues:

🟢 Low-Risk

3 low-risk code issues found

🟡 Medium-Risk

1 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.

Unchecked transfer

The return value of an external transfer/transferFrom call is not checked.

```
function recoverToken(address tokenAddress, uint256 tokenAmount) public virtual onlyOwner {
    IERC20(tokenAddress).transfer(owner(), tokenAmount);
}
```

Recommendation

Use `SafeERC20`, or ensure that the transfer/transferFrom return value is checked.

Exploit scenario

```
contract Token {
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success);
}
contract MyBank{
    mapping(address => uint) balances;
    Token token;
    function deposit(uint amount) public{
        token.transferFrom(msg.sender, address(this), amount);
        balances[msg.sender] += amount;
    }
}
```

Several tokens do not revert in case of failure and return false. If one of these tokens is used in `MyBank`, `deposit` will not revert if the transfer fails, and an attacker can call `deposit` for free..

🟢 **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.

```
string private __identifier;
```

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.

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

Redundant Statements

Detect the usage of redundant statements that have no effect.

```
function _msgData() internal view virtual returns (bytes calldata) {
    this; // silence state mutability warning without generating bytecode - see https://github.com/e
    return msg.data;
}
```

Recommendation

Remove redundant statements if they congest code but offer no value.

Exploit scenario

```
contract RedundantStatementsContract {

    constructor() public {
        uint; // Elementary Type Name
        bool; // Elementary Type Name
        RedundantStatementsContract; // Identifier
    }

    function test() public returns (uint) {
        uint; // Elementary Type Name
        assert; // Identifier
        test; // Identifier
        return 777;
    }
}
```

Each commented line references types/identifiers, but performs no action with them, so no code will be generated for such statements and they can be removed.

🟡 **Medium-Risk:** Should be fixed, could bring problems.

Owner can mint new tokens

```
function mint(address account, uint256 amount) external canMint {
    _mint(account, amount);
}
```

Recommendation

No recommendation

Owner privileges

- 🟢 Owner cannot set fees higher than 25%
- 🟢 Owner cannot pause trading
- 🟢 Owner cannot change max transaction amount
- 🔴 Owner can mint new tokens

⚠️ Owner can send tokens within the contract to his wallet

Extra notes by the team

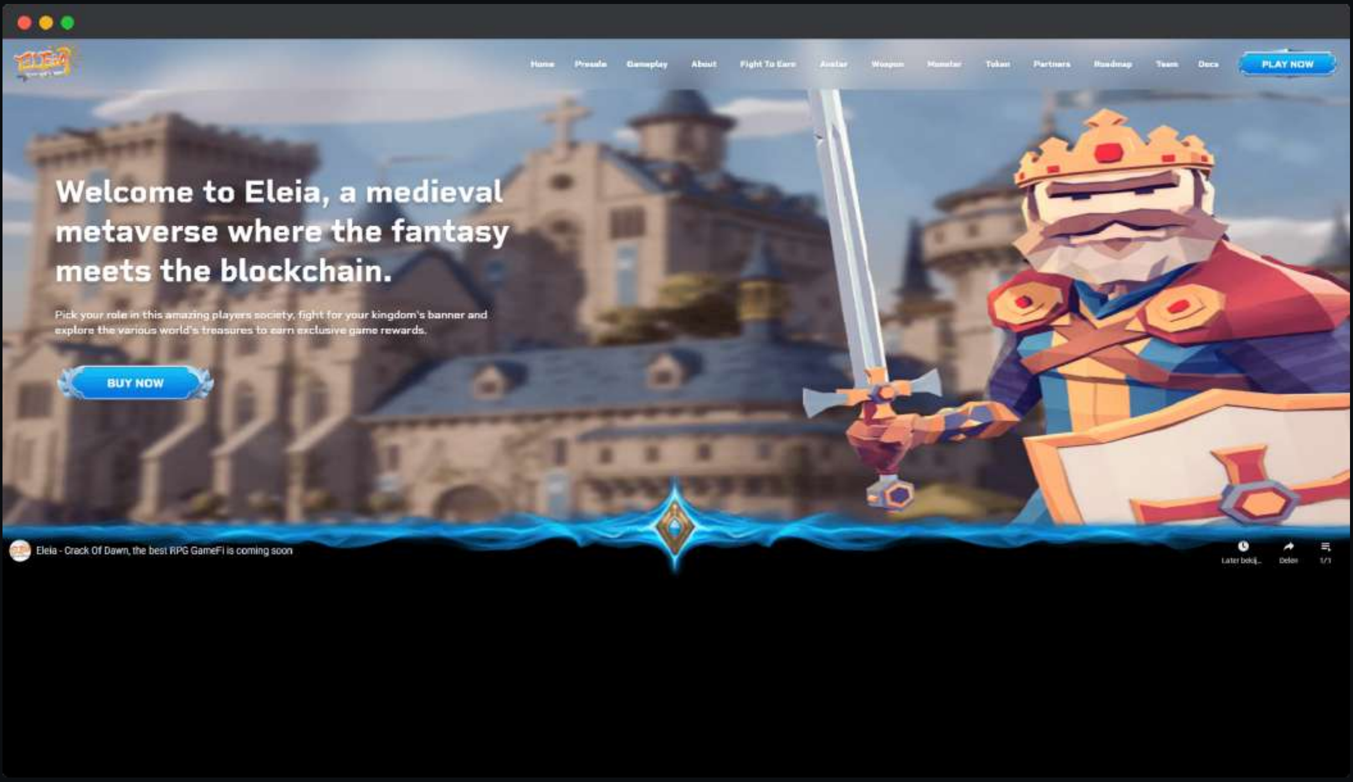
No notes

Contract Snapshot

```
abstract contract MetacryptHelper {
    address private __target;
    string private __identifier;

    constructor(string memory __metacrypt_id, address __metacrypt_target) payable {
        __target = __metacrypt_target;
        __identifier = __metacrypt_id;
        payable(__metacrypt_target).transfer(msg.value);
    }

    function createdByMetacrypt() public pure returns (bool) {
        return true;
    }

    function getIdentifier() public view returns (string memory) {
        return __identifier;
    }
}
```

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.



Welcome to Eleia, a medieval metaverse where the fantasy meets the blockchain.

Pick your role in this amazing players society, fight for your kingdom's banner and explore the various world's treasures to earn exclusive game rewards.

Eleia - Crack Of Dawn, the best RPG GameFi is coming soon

● Mobile Friendly

● Does not contain jQuery errors

● SSL Secured

● No major spelling errors

Project Overview

● Not KYC verified by Coinsult