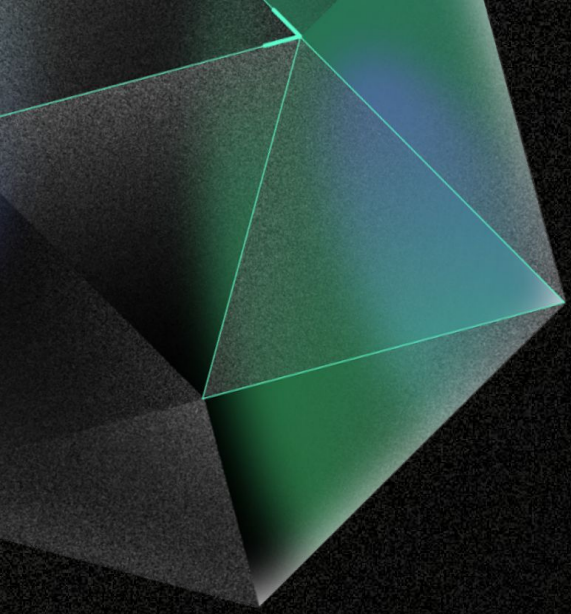
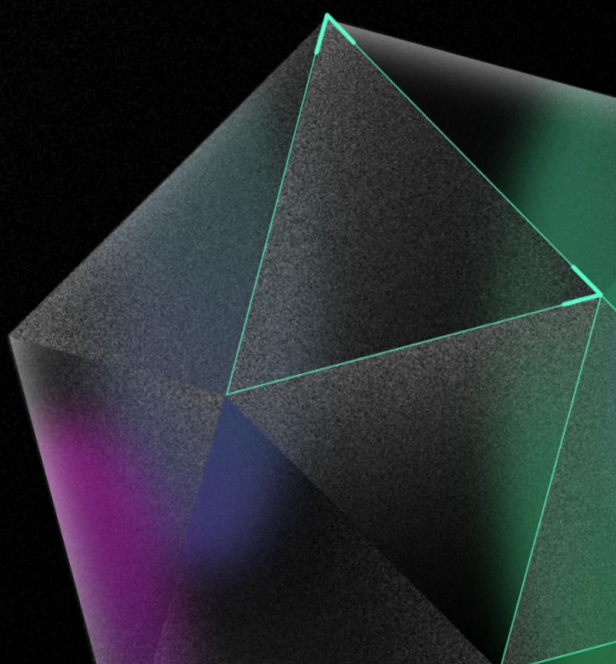


hexens

SMART CONTRACT AUDIT REPORT FOR TRUPENNY TOKEN

30.01.2022



Contents

Summary

During the audit our specialists found 2 critical risk, 1 high risk and multiple informational weaknesses.

Severity	Number of Findings
Critical	2
High	1
Medium	0
Low	0
Informational	2

Total: 5

Scope

The analyzed contracts were located in contracts.zip archive with the following SHA512 checksum:
b40c7f58098447d543cc691b34e401fd3a31570da7bfdb5d5a506a8f6d832762
3f159dec18d5cc9f439450933241ba577d1424f3662cbc0814dafbc919d6459a

Weaknesses

This section contains the list of discovered weaknesses.

1. Missing Authorization

Severity: Critical

Resolution: add onlyOwner or onlySecurityContract modifier

Description:

The **executeOperation()** function on line 187 in the file BotContract.sol is not an authorized function. This gives an opportunity for any attacker to execute swap operations.

```
function executeOperation(
    address[] calldata assets,
    uint256[] calldata amounts,
    uint256[] calldata premiums,
    address,
    bytes calldata params
  )
```

2. Missing Authorization

Severity: Critical

Resolution: add onlyOwner or onlySecurityContract modifier

Description:

The **withdrawERC20()** function on line 220 in the file BotContract.sol is not an authorized function. This gives an opportunity for any attacker to withdraw all tokens from the contract.

```
function withdrawERC20(IERC20 tokenAddress, uint256 amount)
public returns (bool) {
    tokenAddress.transfer(msg.sender, amount);
    return true;
  }
```

3. Signature Replay Attack

Severity: High

Resolution: Implement EIP-712 compliant signature and nonce mechanism to verify that a signature cannot be replayed both in and out of the current domain

Description:

The **recoverSigner()** function on line 89 in the file SecurityContract.sol verifies a signature, but does not consider the cross-domain replays and also no nonce is used, which gives opportunity for replay attacks.

```
function recoverSigner(
    address asset,
    uint256 initialAmount,
    uint256 expectedAmount,
    uint256 slippage,
    uint256 maxGasPrice,
    bool isFlashloan,
    bytes memory params,
    bytes memory signature
  )
```

4. Unused Function

Severity: Informational

Resolution: consider removing the function

Description:

The **recoverSigner()** function on line 229 in the file BotContract.sol is not in use, and seems to be a duplicate of SecurityContract.sol.

```
function recoverSigner(
    address asset,
...
    bytes memory signature
```

5. Misprints in the Contract

Severity: Informational

Resolution: fix the misprints

Description:

Throughout the contract the function name **_callUniswapLikeFunctions()** is misspelled and written as **_callUniswapLikeFuncions()** instead.

```
function _callUniswapLikeFuncions(
...
lastTokenInfo = _callUniswapLikeFuncions(UNISWAP_V2_ROUTER_02,
path, asset, amount);
...
lastTokenInfo = _callUniswapLikeFuncions(SUSHI_V2_ROUTER_02,
path, asset, amount);
```


hexens