



COGNITOS

The next generation of blockchain security



Skyren

Gold Audit

Deep Scan Mode Screening

November, 29
2024

Disclaimer

Cognitos provides due-diligence project audits for various projects. Cognitos in no way guarantees that a project will not remove liquidity, sell off teamsupply, or otherwise exit scam.

Cognitos does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to an audit in no way guarantees that a team will not remove all liquidity ("Rug Pull"), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

Cognitos in no way takes responsibility for any losses, nor does Cognitos encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Cognitos does not endorse, recommend, support, or suggest any projects that have been audited. An audit is an informational report based on our findings, We BEP recommend you do your own research, we will never endorse any project to invest in.

The badge of Audit, KYC, Vetted and Safu is not a guarantee for safety. your reliance on a badge is solely at your own risk. we are not responsible for your investment loss and hereby expressly disclaim any liabilities that may arise from your use or reference of the badge.

Table of content

| | |
|---------------------------|----|
| Disclaimer | 1 |
| Table of Content | 2 |
| Audit Scope | 3 |
| • Project Overview | 4 |
| • Token Data | 5 |
| • Security Detection | 6 |
| • Vulnerability Summary | 7 |
| • Vulnerability Scan | 8 |
| Locked Ether | |
| Public Burn | |
| Reentrancy | |
| • Weakness Classification | 12 |
| • Website Profiling | 14 |
| • Team Profiling | 18 |

Audit Scope

Cognitos was commissioned by Skyren to perform an audit based on the following code:

<https://polygonscan.com/address/0xBa74014e2A8ab23b14f7D6d067494A0Bf1567bB2#code>

Note that we only audited the code available to us on this URL at the time of the audit. If the URL is not from any block explorer (main net), it may be subject to change. Always check the contract address on this audit report and compare it to the token you are doing research for.

Audit Method

Cognitos's manual smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. This process is conducted to discover errors, issues and security vulnerabilities in the code in order to suggest improvements and ways to fix them.

Automated Vulnerability Check

Cognitos uses software that checks for common vulnerability issues within smart contracts. We use automated tools that scan the contract for security vulnerabilities such as integer-overflow, integer-underflow, out-of-gas-situations, unchecked transfers, etc.

Manual Code Review

Cognitos's manual code review involves a human looking at source code, line by line, to find vulnerabilities. Manual code review helps to clarify the context of coding decisions. Automated tools are faster but they cannot take the developer's intentions and general business logic into consideration.



Project Overview

Name & Logo



Skyren

Project Statement

Your Gateway to Exclusive Cryptocurrency Airdrops
Skyren is a groundbreaking airdrop collection service that connects cryptocurrency enthusiasts to unique token airdrops they might have missed or were unaware of their eligibility. With cutting-edge proprietary technology, Skyren tirelessly scans all layer one, two, and standalone blockchains to unearth new and exciting airdrops, ensuring its users never miss out on potential opportunities.

By holding the \$SKYRN token, users can enjoy the benefits of cryptocurrency airdrops without the need to search for projects and become eligible themselves. Skyren simplifies the complex task of airdrop hunting with a user-friendly interface, offering a streamlined experience accessible to anyone.

Website & Social Media

- Website
- Telegram
- Twitter
- Instagram
- Medium
- Gitbook

skyren.io

<https://t.me/SkyrenDAO>

https://x.com/Skyren_Official

<https://www.instagram.com/skyrendao/>

<https://skyren.medium.com/>

<https://skyren-foundation.gitbook.io/skyren-technical-white-paper>

Blockchain

- Network
- Contract

Polygon

0xBa74014e2A8ab23b14f7D6d067494A0Bf1567bB2 (verified)



Token Data

**Token
Symbol**

SKYRN

Token Name

Skyren

**Contract
Address**

0xBa74014e2A8ab23b14f7D6d067494A0Bf1567bB2

**Compiler
Version**

v0.8.17+commit.8df45f5f

Total Supply

190,000,000 SKYRN

Decimals

18

**Contract
Creator**

0x5e5836499ee360331a2bdee2b60ffa261ed22418

**Contract
Owner**

0x5e5836499ee360331a2bdee2b60ffa261ed22418



Security Detection

Risky Item

0

Attention Item

3

Safe Attention Risky

Contract Security

| | | |
|------------------------------|--|-----|
| Contract Verified | | Yes |
| Proxy Contract | | No |
| Mint Function | | No |
| Retrieves Ownership Function | | No |
| Authority to Change Balance | | No |
| Hidden Owner | | No |
| Self-destruct Function | | No |
| External Call Risk | | No |

Honeypot Risk

| | | |
|--------------------------------|--|-----|
| Appear to be a Honeypot | | No |
| Can be Bought | | Yes |
| Trading Cooldown Function | | Yes |
| Anti_whale Function | | Yes |
| Tax Modified Function | | Yes |
| Blacklist Function | | Yes |
| Whitelist Function | | Yes |
| Personal Addresses Tax Changes | | No |

Vulnerability Summary

Total Findings



- 0 Critical
- 0 Major
- 4 Medium
- 6 Minor
- 8 Informational

Severity

• Critical

-

• Major

-

• Medium

Locked Ether
Public Burn
Reentrancy
Incorrect Access Control

• Minor

Precision Loss During Division By Large Numbers
Using Extcodesize To Check For Externally Owned Accounts
Unchecked Array Length
Outdated Compiler Version
Function Returns Type And No Return
Event Based Reentrancy

• Info

Block Values As A Proxy For Time
In-line Assembly Detected
Require With Empty Message
Boolean Equality
Missing Indexed Keywords In Events
Unused Receive Fallback
Missing Inheritance
Return Inside Loop

Vulnerability Scan

Locked Ether

| | |
|----------------------|---------|
| Severity | Medium |
| Confidence Parameter | Certain |

Vulnerability Description

The smart contract is accepting Ether at its address. This ether can be stored but due to misconfigurations or missing functions, there is no way to transfer this Ether out of the contract's address. This causes the Ether to be locked inside the contract.

Scanning Line:

```
294 contract BuyBackWallet is Ownable{
1677 contract LPWallet is Ownable{
4292 contract TaxHelperCamelotV2 is Ownable{
4407 contract TaxHelperUniswapV2 is Ownable{
```

Recommendation:

Allow the contract to have a function to withdraw Ether out of the contract, either to some externally owned account or another contract.

Vulnerability Scan

PUBLIC BURN

| | |
|----------------------|---------|
| Severity | Medium |
| Confidence Parameter | Certain |

Vulnerability Description

The contract was found to be using public or an external burn function. The function was missing access control to prevent another user from burning their tokens. Also, the burn function was found to be using a different address than `msg.sender`.

Scanning Line:

```

3423 function burn(uint256 amount) public {
3424     address taxHelper = IMintFactory(s.factory).getTaxHelperAddress(s.taxHelperIndex);
3425     require(msg.sender == taxHelper || msg.sender == owner(), "RA");
3426     _burn(owner(), amount);
3427 }
```

Recommendation:

Consider adding access control modifiers to the burn function to prevent unauthorized users from burning tokens. Use the `onlyOwner` modifier from the OpenZeppelin Ownable contract to restrict access. The burn function should also use `msg.sender` in the `_from` argument to ensure that only the owner can call the function.

Vulnerability Scan

REENTRANCY

Severity Medium
Confidence Parameter Tentative

Vulnerability Description

In a Re-entrancy attack, a malicious contract calls back into the calling contract before the first invocation of the function is finished. This may cause the different invocations of the function to interact in undesirable ways, especially in cases where the function is updating state variables after the external calls.

This may lead to loss of funds, improper value updates, token loss, etc.

Scanning Line:

```

325-333      function sendEthToTaxHelper() external returns (uint256) {
1707-1715    function sendEthToTaxHelper() external returns (uint256) {
2152-2241    function handleTaxes(address sender, address recipient, uint256 amount) public
virtual returns (uint256 totalTaxAmount) {
3985-4000    function transfer(address recipient, uint256 amount) public returns (bool) {
4011-4032    function transferFrom(address sender, address recipient, uint256 amount) public
returns (bool) {
4197-4205    function createToken (
4316-4341    function initiateBuyBackTax(address _token, address _wallet) payable external
isToken returns(bool) {
4343-4375    function initiateLPTokenTax(address _token, address _wallet) external isToken
returns (bool) {
    
```

Recommendation:

It is recommended to add a [Re-entrancy Guard] to the functions making external calls. The functions should use a Checks-Effects-Interactions pattern. The external calls should be executed at the end of the function and all the state-changing must happen before the call.

Weakness Classification

| | AI Scan | Human Review | Result | |
|---------|--------------------------------------|--------------|--------|--------|
| CTS 000 | Function Default Visibility | ✓ | ✓ | Passed |
| CTS 001 | Integer Overflow and Underflow | ✓ | ✓ | Passed |
| CTS 002 | Outdated Compiler Version | ✓ | ✓ | Passed |
| CTS 003 | Floating Pragma | ✓ | ✓ | Passed |
| CTS 004 | Unchecked Call Return Value | ✓ | ✓ | Passed |
| CTS 005 | Unprotected Ether Withdrawal | ✓ | ✓ | Passed |
| CTS 006 | Unprotected SELFDESTRUCT Instruction | ✓ | ✓ | Passed |
| CTS 007 | Reentrancy | ✓ | ✓ | Passed |
| CTS 008 | State Variable Default Visibility | ✓ | ✓ | Passed |
| CTS 009 | Uninitialized Storage Pointer | ✓ | ✓ | Passed |
| CTS 010 | Assert Violation | ✓ | ✓ | Passed |
| CTS 011 | Use of Deprecated Solidity Functions | ✓ | ✓ | Passed |
| CTS 012 | Delegatecall to Untrusted Callee | ✓ | ✓ | Passed |
| CTS 013 | DoS with Failed Call | ✓ | ✓ | Passed |
| CTS 014 | Transaction Order Dependence | ✓ | ✓ | Passed |
| CTS 015 | Authorization through tx.origin | ✓ | ✓ | Passed |
| CTS 016 | Block values as a proxy for time | ✓ | ✓ | Passed |
| CTS 017 | Signature Malleability | ✓ | ✓ | Passed |
| CTS 018 | Incorrect Constructor Name | ✓ | ✓ | Passed |

| | | AI Scan | Human Review | Result |
|---------|---|---------|--------------|--------|
| CTS 019 | Shadowing State Variables | ✓ | ✓ | Passed |
| CTS 020 | Weak Sources of Randomness from Chain Attributes | ✓ | ✓ | Passed |
| CTS 021 | Missing Protection against Signature Replay Attacks | ✓ | ✓ | Passed |
| CTS 022 | Lack of Proper Signature Verification | ✓ | ✓ | Passed |
| CTS 023 | Requirement Violation | ✓ | ✓ | Passed |
| CTS 024 | Write to Arbitrary Storage Location | ✓ | ✓ | Passed |
| CTS 025 | Incorrect Inheritance Order | ✓ | ✓ | Passed |
| CTS 026 | Insufficient Gas Griefing | ✓ | ✓ | Passed |
| CTS 027 | Arbitrary Jump with Function Type Variable | ✓ | ✓ | Passed |
| CTS 028 | DoS With Block Gas Limit | ✓ | ✓ | Passed |
| CTS 029 | Typographical Error | ✓ | ✓ | Passed |
| CTS 030 | Right-To-Left-Override control character (U+202E) | ✓ | ✓ | Passed |
| CTS 031 | Presence of unused variables | ✓ | ✓ | Passed |
| CTS 032 | Unexpected Ether balance | ✓ | ✓ | Passed |
| CTS 033 | Hash Collisions With Multiple Variable Length Arguments | ✓ | ✓ | Passed |
| CTS 034 | Message call with hardcoded gas amount | ✓ | ✓ | Passed |
| CTS 035 | Code With No Effects | ✓ | ✓ | Passed |
| CTS 036 | Unencrypted Private Data On-Chain | ✓ | ✓ | Passed |



**Security
Detection**

Website Security



Minimal Low Security Risk Medium High Critical

Our automated scan did not detect malware on your site.

**Sitescan
Report**

| | |
|------------------------------|---|
| Normalized URL | - |
| Submission date | - |
| Server IP address | - |
| Country | - |
| Web Server | - |
| Malicious files | - |
| Suspicious files | - |
| Potentially Suspicious files | - |
| Clean files | - |
| External links detected | - |
| Iframes scanned | - |
| Blacklisted | - |

**Scanned
files analysis**

| | |
|------------------------------|---|
| Malicious files | - |
| Suspicious files | - |
| Potentially Suspicious files | - |
| Clean files | - |

**Malware
Checked**



**Blacklist
Checked**



SSL Checked



Server

-

Chain 1

-

**Technology
Profiler**



Dev & Team Informations

Team Data

We don't find developer and team information on their website

Cognitos Project Audit has been completed for **Skyren - POLYGON**

Block number : 00000232



**This result is only valid if viewed on
www.cognitos.io**



 **COGNITOS**

The next generation of blockchain security