AudiTBlock

LIGHTSWAPRouter

v0.7.6+commit.7338295f

Low-Risk	Medium-Risk	High-Risk
low-risk code	medium-risk code	high-risk code

LIGHTSWAPRouter

Contract Deployed On nova.arbiscan.io

0xefaedbde098bb88b36646b03765503c72ff1f5b9

Disclaimer

AudiTBlock 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.

AudiTBlock is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your ownresearch.

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

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

Tokenomics

- Arbitrum Nova

Source Code

- AudiTBlock was complete audit phases to perform an audit based on the following smart contract:

- https://nova.arbiscan.io/address/0xefaedbde098bb88b36646b03765503c72ff1f5b9#code

LIGHTSWAPRouter.removeLiquidity(address,address,uint256,uint256,uint256,address,uint256) (contracts/contract.sol#503-533) ignores return value by

ILIGHTSWAPPair(pair).transferFrom(msg.sender,pair,liquidity) (contracts/contract.sol#519)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

LIGHTSWAPRouter._swapSupportingFeeOnTransferTokens(address[],address).i (contracts/contract.sol#894) is a local variable never initialized

LIGHTSWAPRouter._swap(uint256[],address[],address).i (contracts/contract.sol#698) is a local variable never initialized

LIGHTSWAPLibrary.getAmountsOut(address,uint256,address[]).i (contracts/contract.sol#1160) is a local variable never initialized

Reference:https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256)

(contracts/contract.sol#392-437) ignores return value by ILIGHTSWAPFactory(factory).createPair(tokenA,tokenB) (contracts/contract.sol#402)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Tested Contract Files

The following are the MD5 hashes of the reviewed files. A file with a different MD5 hash has been modified, intentionally or otherwise,
after the security review. You are cautioned that a different MD5 hash could be (but is not necessarily) an indication of a changed
condition or potential vulnerability that was not within the scope of the review

File	Fingerprint (MD5
Contracts/LIGHTSWAPRouter.sol	a3b86986d6558032bcf6022ac36a02754

Used Code from other Frameworks/Smart Contracts (direct imports)

Dependency / Import Path	Source Sha1 Hash
Contracts/library, interfaces	a389efa78632472148e2e1a118bd590ba975ef18

Snapshot 0.2

```
LIGHTSWAPRouter.constructor(address,address)._factory (contracts/contract.sol#382)
lacks a zero-check on :
        - factory = _factory (contracts/contract.sol#383)
LIGHTSWAPRouter.constructor(address,address)._WETH (contracts/contract.sol#382)
lacks a zero-check on :
        - WETH = _WETH (contracts/contract.sol#384)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-
address-validation

TransferHelper.safeApprove(address,address,uint256) (contracts/contract.sol#1192-
1201) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version=0.6.6 (contracts/contract.sol#5) allows old versions
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity

Low level call in TransferHelper.safeApprove(address,address,uint256)
(contracts/contract.sol#1192-1201):
    - (success,data) = token.call(abi.encodeWithSelector(0x095ea7b3,to,value))
(contracts/contract.sol#1194-1196)
Low level call in TransferHelper.safeTransfer(address,address,uint256)
(contracts/contract.sol#1203-1212):
    - (success,data) = token.call(abi.encodeWithSelector(0xa9059cbb,to,value))
(contracts/contract.sol#1205-1207)
Low level call in TransferHelper.safeTransferFrom(address,address,address,uint256)
(contracts/contract.sol#1214-1228):
    - (success,data) = token.call(abi.encodeWithSelector(0x23b872dd,from,to,value))
(contracts/contract.sol#1221-1223)
Low level call in TransferHelper.safeTransferETH(address,uint256)
(contracts/contract.sol#1230-1233):
    - (success) = to.call{value: value}(new bytes(0)) (contracts/contract.sol#1231)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function ILIGHTSWAPPair.DOMAIN_SEPARATOR() (contracts/contract.sol#67) is not
in mixedCase
Function ILIGHTSWAPPair.PERMIT_TYPEHASH() (contracts/contract.sol#69) is not in
mixedCase
Function ILIGHTSWAPPair.MINIMUM_LIQUIDITY() (contracts/contract.sol#100) is not
in mixedCase
Function ILIGHTSWAPRouter01.WETH() (contracts/contract.sol#140) is not in
mixedCase
Variable LIGHTSWAPRouter.WETH (contracts/contract.sol#375) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-
to-solidity-naming-conventions
```

Snapshot 0.3

```
 Variable
ILIGHTSWAPRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountADesired (contracts/contract.sol#145) is too similar
to
ILIGHTSWAPRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountBDesired (contracts/contract.sol#146)
Variable
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountADesired (contracts/contract.sol#442) is too similar to
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountBDesired (contracts/contract.sol#443)
Variable
LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256).a
mountADesired (contracts/contract.sol#395) is too similar to
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountBDesired (contracts/contract.sol#443)
Variable
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountADesired (contracts/contract.sol#442) is too similar to
ILIGHTSWAPRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountBDesired (contracts/contract.sol#146)
Variable
LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256).a
mountADesired (contracts/contract.sol#395) is too similar to
ILIGHTSWAPRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountBDesired (contracts/contract.sol#146)
Variable
LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256).a
mountADesired (contracts/contract.sol#395) is too similar to
LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256).a
mountBDesired (contracts/contract.sol#396)
Variable
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountADesired (contracts/contract.sol#442) is too similar to
LIGHTSWAPRouter._addLiquidity(address,address,uint256,uint256,uint256,uint256).a
mountBDesired (contracts/contract.sol#396)
Variable
ILIGHTSWAPRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,a
ddress,uint256).amountADesired (contracts/contract.sol#145) is too similar
to
LIGHTSWAPRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,addr
ess,uint256).amountBDesired (contracts/contract.sol#443)
Variable
```

Auto

```
235: address public immutable override factory;
 236:  address public immutable override WETH;
243:  constructor(address _factory, address _WETH) public {
        factory = _factory;
        WETH = _WETH;
    }
        372:  uint value = approveMax ? uint(-1) : liquidity;

693: pair = address(uint(keccak256(abi.encodePacked(
                hex'ff',
                factory,
                keccak256(abi.encodePacked(token0, token1)),
                hex'540c9fbac3bf3431814da396bfc770ec61400eb56cccf36110c32b07ecc491b3' //
init code hash
            ))));
    }
```

Constant/View/Pure

Overriding public state variable changes state mutability from "pure" to "view".
address public immutable override factory;
Overriding public state variable changes state mutability from "pure" to "view".

Similar variable names

TypeError: Explicit type conversion not allowed from "uint256" to "address".
697 | pair = address(uint(keccak256(abi.encodePacked(

No return

Explicit type conversion not allowed from "int_const -1" to "uint256".

0.2 SOLIDITY UNIT TESTING

Progress: Starting
PASS ✓ ✓ **Tested**

✓ Check winning proposal
✓ Check winnin proposal with return value
✓ Before all
✓ Check success
✓ Check success2
✓ Check sender and value

Result for tests Passed:

0Time Taken: 0.54s

```
library SafeMath {
    function add(uint x, uint y) internal pure returns (uint z) {
        require((z = x + y) >= x, 'ds-math-add-overflow');
    }

    function sub(uint x, uint y) internal pure returns (uint z) {
        require((z = x - y) <= x, 'ds-math-sub-underflow');
    }

    function mul(uint x, uint y) internal pure returns (uint z) {
        require(y == 0 || (z = x * y) / y == x, 'ds-math-mul-overflow');
    }
}

library LIGHTSWAPLibrary {
    using SafeMath for uint;

    // returns sorted token addresses, used to handle return values from pairs sorted in this order
    function sortTokens(address tokenA, address tokenB) internal pure returns (address token0,
address token1) {
        require(tokenA != tokenB, 'LIGHTSWAPLibrary: IDENTICAL_ADDRESSES');
        (token0, token1) = tokenA < tokenB ? (tokenA, tokenB) : (tokenB, tokenA);
        require(token0 != address(0), 'LIGHTSWAPLibrary: ZERO_ADDRESS');
    }

    // calculates the CREATE2 address for a pair without making any external calls
    function pairFor(address factory, address tokenA, address tokenB) internal pure returns (address
pair) {
        (address token0, address token1) = sortTokens(tokenA, tokenB);
        pair = address(uint(keccak256(abi.encodePacked(
                hex'ff',
```

```
contract2.sol:40:5: Error: Function name must be in mixedCase
contract2.sol:57:5: Error: Function name must be in mixedCase
contract2.sol:77:5: Error: Function name must be in mixedCase
contract2.sol:236:39: Error: Variable name must be in mixedCase
contract2.sol:239:29: Error: Avoid to make time-based decisions in your business logic
contract2.sol:239:46: Error: Use double quotes for string literals
contract2.sol:243:35: Error: Variable name must be in mixedCase
contract2.sol:271:55: Error: Use double quotes for string literals
contract2.sol:276:55: Error: Use double quotes for string literals
contract2.sol:765:45: Error: Avoid to use low level calls.
contract2.sol:766:76: Error: Use double quotes for string literals
contract2.sol:771:45: Error: Avoid to use low level calls.
contract2.sol:772:76: Error: Use double quotes for string literals
contract2.sol:777:45: Error: Avoid to use low level calls.
contract2.sol:778:76: Error: Use double quotes for string literals
contract2.sol:782:27: Error: Avoid to use low level calls.
contract2.sol:783:26: Error: Use double quotes for string literal
```

Manual and Automated Vulnerability Test

CRITICAL ISSUES

During the audit, AudiTBlock experts found **4 medium Critical issues** in the code of the smart contract.

HIGH ISSUES

During the audit, AudiTBlock experts found **0 High issues** in the code of the smart contract.

MEDIUM ISSUES

During the audit, AudiTBlock experts found **2 Medium issues** in the code of the smart contract.

LOW ISSUES

During the audit, AudiTBlock experts found **6 Low issues** in the code of the smart contract.

INFORMATIONAL ISSUES

During the audit, AuditBlock experts found **2 Informational issues** in the code of the smart contract.

SWC Attacks

ID	Title		Test Result
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	✔
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	✔
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	✔
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	✔
SWC-127	Arbitrary Jump with Function TypeVariable	CWE-695: Use of Low-Level Functionality	✔
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	✔
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	✔
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	✔

ID	Title		Test Result
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	✔
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	✔
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	✔
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	✔
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	✔
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	✔
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	✔
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	✔
SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	✔
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	✔

Owner privileges

- Verify Claims
- The contract block difficulty 540000325454446
  Status: tested and verified ✓
- Status: tested 1 and verified ✓
- Status: tested 2 and verified ✓
- Status: tested 3 and verified ✓
- Status: tested 4 and verified ✓
- Status: tested and verified ✓

Executive Summary

Two (2) independent AuditBlock experts performed an unbiased and isolated audit of the smart contract. The final debriefs

The overall code quality is good and not overloaded with unnecessary functions, these is greatly

benefiting the security of the contract. It correctly implemented widely used and reviewed contracts he main goal of the audit was to verify the claims regarding the security of the smart contract and the claims inside the scope of work.

During the audit, no issues were found after the manual and automated security testing.

Deployed On Nova Arbiscan

VERIFIED ✔

https://nova.arbiscan.io/address/0xefaedbde098bb88b36646b03765503c72ff1f5b9#code