ANALYTIX AUDIT

28/09/2022

PREMIUM AUDIT

NAME OF PROJECT	MagicLink
TOKEN BEP20 ADDRESS	0xA3Cb82a535c0d9066Ba5160a0D4cc6350b88ed61
WEBSITE / TELEGRAM	https://magiclinktoken.com/

DISCLAIMER: PLEASE READ FULL AUDIT

IMPORTANT DISCLAIMER

I.	**NOT RESPONSIBLE**	Analytix Audit holds no responsibility for any actions from the project in this audit.
II.	**NOT GUARANTEE**	Analytix Audit in no way guarantees that a project will not remove liquidity, sell off team tokens, or exit scam.
III.	**INFORMATION**	Analytix Audit researches and provides public information about the project in an easy-to-understand format for the common person.
IV.	**AUDIT AGREEMENT**	This audit agreement does not guarantee ANY illicit actions by the project team and does not serve as an advocation for the project.
V.	**NO ADVICE**	Analytix Audit in no way takes responsibility for any losses, not does Analytix Audit encourage any speculative investments.
VI.	**DISCLAIMER**	The information provided in this audit is for information purposes only and should not be considered investment advice. Analytix Audit does not endorse, recommend, support, or suggest any projects that have been audited.

TOKENOMICS - OVERVIEW

MagicLink Token Holders

Home / MagicLink / Token Holders Chart

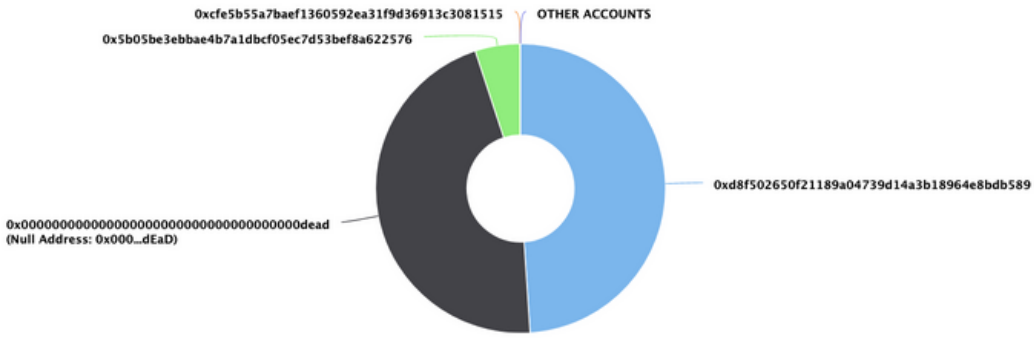
Range: Top 100

The top 100 holders collectively own 100.00% (200,000,000.00 Tokens) of MagicLink

Token Total Supply: 200,000,000.00 Token | Total Token Holders: 4

MagicLink Top 100 Token Holders

Source: BscScan.com



(A total of 200,000,000.00 tokens held by the top 100 accounts from the total supply of 200,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xd8f502650f21189a04739d14a3b18964e8bdb589	97,902,000	48.9510%
2	Null Address: 0x000...dEaD	92,000,000	46.0000%
3	0x5b05be3ebbae4b7a1dbcf05ec7d53bef8a622576	10,000,000	5.0000%
4	0xcfe5b55a7baef1360592ea31f9d36913c3081515	98,000	0.0490%

More than 15% tokens are unlocked

(This comment refers to when the audit was carried out maybe team locked tokens later)

CONTRACT CODE - PRIVILEGES

```
520     * Requirements:
521     *
522     * - the calling contract must have an ETH balance of at least `value`.
523     * - the called Solidity function must be `payable`.
524     *
525     * _Available since v3.1._
526     */
527    function functionCallWithValue(
528        address target,
529        bytes memory data,
530        uint256 value
531    ) internal returns (bytes memory) {
532        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
533    }
534
535    /**
536     * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
537     * with `errorMessage` as a fallback revert reason when `target` reverts.
538     *
539     * _Available since v3.1._
540     */
541    function functionCallWithValue(
542        address target,
543        bytes memory data,
544        uint256 value,
545        string memory errorMessage
546    ) internal returns (bytes memory) {
547        require(address(this).balance >= value, "Address: insufficient balance for call");
548        require(isContract(target), "Address: call to non-contract");
549
550        (bool success, bytes memory returndata) = target.call{value: value}(data);
```

- **Mint:** Owner cannot mint new tokens
- **Fees:** Owner can't change fees
- **Trading:** Owner can't change trading status
- **Max Tx:** Owner can't change maxTx
- **Max Wallet:** Owner can't change max wallet
- **Blacklist:** Owner can't blacklist wallet

Informational*

CONTRACT CODE - SNIPPETS

Informational: AntiBot measurements applied in the first 20 blocks (60 seconds)

```
    if (block.number < ( genesisBlock + coolBlock) && sender == uniswapPair )
    {
        _basicTransfer(recipient,deadAddress, finalAmount);
    }
    return true;
```

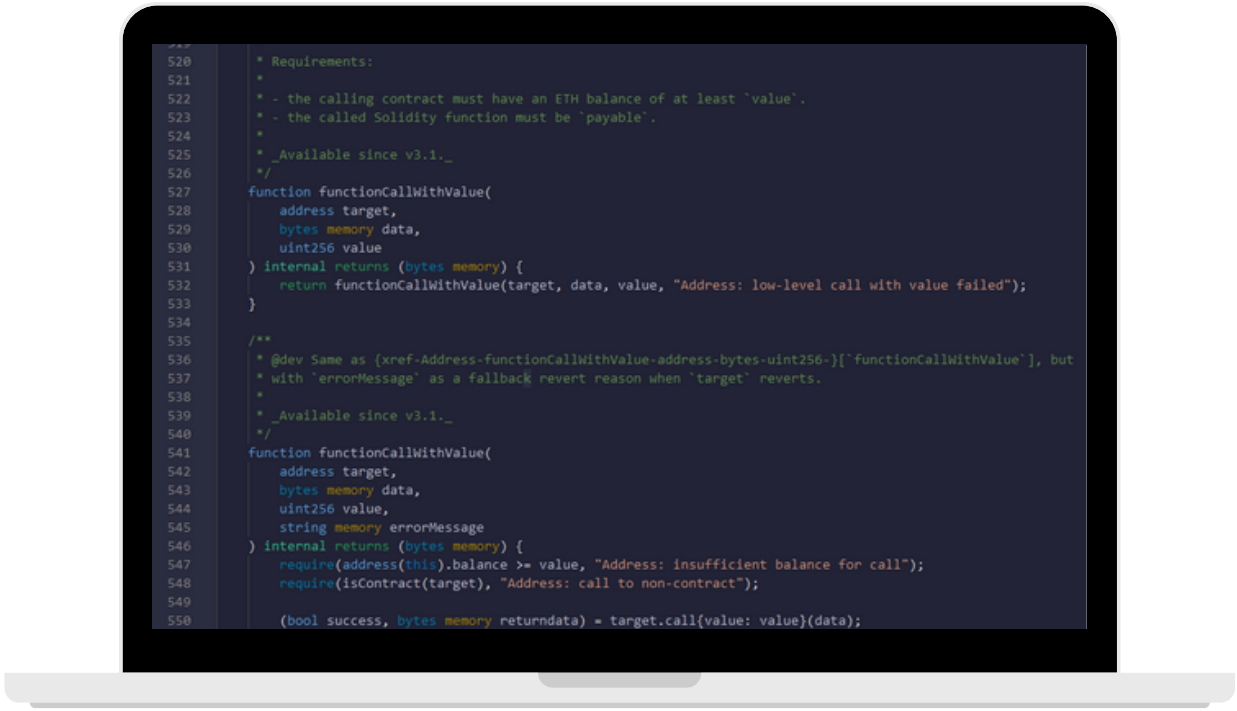
Each bot buying in the first 60 seconds after launch will automatically transfer to a dead wallet.

Informational: Safemath

```
library SafeMath {

    function add(uint256 a, uint256 b) internal pure returns (uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");

        return c;
    }

    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        return sub(a, b, "SafeMath: subtraction overflow");
    }

    function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b <= a, errorMessage);
        uint256 c = a - b;

        return c;
    }

    function mul(uint256 a, uint256 b) internal pure returns (uint256) {
        if (a == 0) {
            return 0;
        }

        uint256 c = a * b;
        require(c / a == b, "SafeMath: multiplication overflow");
```

As compiler version 0. 8.17 safemath can the use be avoided

Owner Privileges:



```
    * Requirements:
    *
    * - the calling contract must have an ETH balance of at least `value`.
    * - the called Solidity function must be `payable`.
    *
    * _Available since v3.1._
    */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
    }

    /**
     * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
     * with `errorMessage` as a fallback revert reason when `target` reverts.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, "Address: insufficient balance for call");
        require(isContract(target), "Address: call to non-contract");

        (bool success, bytes memory returndata) = target.call{value: value}(data);
```

function renounceOwnership() public virtual onlyOwner {
function transferOwnership(address newOwner) public virtual onlyOwner {
function setMarketPairStatus(address account, bool newValue) public onlyOwner {
function setIsTxLimitExempt(address holder, bool exempt) external onlyOwner {
function setIsExcludedFromFee(address account, bool newValue) public onlyOwner {

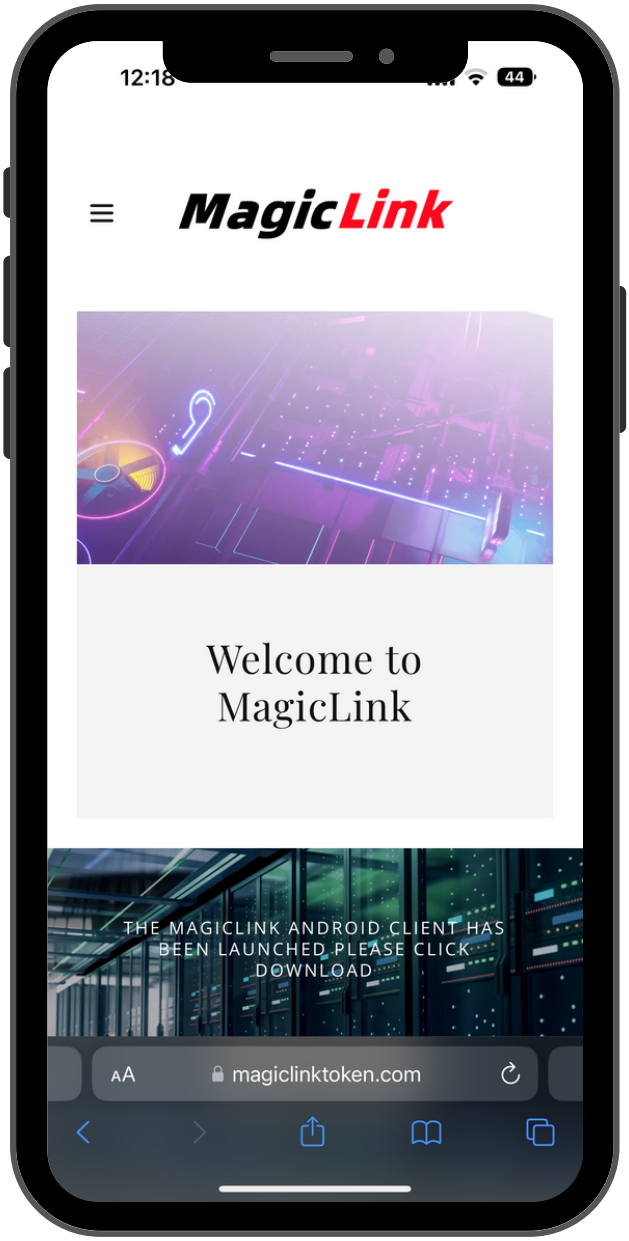
Issue Checking:

N°	Issue description.	Checking Status.
1	Compiler Errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module	Passed
21	Fallback function securitv.	Passed

Audit Result:

PASSED

WEBSITE - OVERVIEW



Domain Registration:

1 year (2023-08-29)

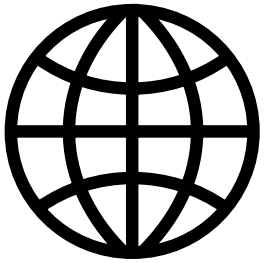
SSL CERTIFICATE: A

(https://www.ssllabs.com)

SOCIAL MEDIA - OVERVIEW

MAGICLINKOFFICIAL

MAGICLINK_



MAGICLINKTOKEN.COM