BlockSAFU

ADVANCE MANUAL SMART CONTRACT AUDIT

Project: Baby Pi Network

Website: babypinetwork.com

✓ PASSED

BlockSAFU Score:

82

Contract Address:

0x2Bb991638681049d76D51A059edaf963239e1C0F

DISCLAMER

BlockSAFU has completed this report to provide a summary of the Smart Contract functions, and any security, dependency, or cybersecurity vulnerabilities. This is often a constrained report on our discoveries based on our investigation and understanding of the current programming versions as of this report's date. To understand the full scope of our analysis, it is vital for you to at the date of this report. To understand the full scope of our analysis, you need to review the complete report. Although we have done our best in conducting our investigation and creating this report, it is vital to note that you should not depend on this report and cannot make any claim against BlockSAFU or its Subsidiaries and Team members on the premise of what has or has not been included in the report. Please remember to conduct your independent examinations before making any investment choices. We do not provide investment advice or in any way claim to determine if the project will be successful or not.

By perusing this report or any portion of it, you concur to the terms of this disclaimer. In the unlikely situation where you do not concur to the terms, you should immediately terminate reading this report, and erase and discard any duplicates of this report downloaded and/or printed by you. This report is given for data purposes as it were and on a non-reliance premise and does not constitute speculation counsel. No one should have any right to depend on the report or its substance, and BlockSAFU and its members (including holding companies, shareholders, backups, representatives, chiefs, officers, and other agents) BlockSAFU and its subsidiaries owe no obligation of care towards you or any other person, nor does BlockSAFU make any guarantee or representation to any individual on the precision or completeness of the report.

ABOUT THE AUDITOR:

BlockSAFU (BSAFU) is an Anti-Scam Token Utility that reviews Smart Contracts and Token information to Identify Rug Pull and Honey Pot scamming activity. BlockSAFUs Development Team consists of several Smart Contract creators, Auditors Developers, and Blockchain experts. BlockSAFU provides solutions, prevents, and hunts down scammers. BSAFU is a utility token with features Audit, KYC, Token Generators, and Bounty Scammers. It will enrich the crypto ecosystem.

OVERVIEW

BlockSAFU was commissioned by Baby Pi Network to complete a Smart Contract audit. The objective of the Audit is to achieve the following:

- Review the Project and experience and Development team
- Ensure that the Smart Contract functions are necessary and operate as intended.
- Identify any vulnerabilities in the Smart Contract code.

DISCLAIMER: This Audit is intended to inform about token Contract Risks, the result does not imply an endorsement or provide financial advice in any way, all investments are made at your own risk. (https://blocksafu.com/)

SMART CONTRACT REVIEW

Token Name	**Baby Pi Network**
Token Symbol	**BPI**
Token Decimal	9
Total Supply	3,140,000,000 BPI
Contract Address	0x2Bb991638681049d76D51A059edaf963239e1C0F
Deployer Address	0xE7A6e9Eb15Ab1C8C0f59f1b0a546788357eEd49f
Owner Address	0xE7A6e9Eb15Ab1C8C0f59f1b0a546788357eEd49f
Tax Fees Buy	0%
Tax Fees Sell	0%
Gas Used for Buy	*will be updated after the DEX listing*
Gas Used for Sell	*will be updated after the DEX listing*
Contract Created	Jul-19-2022 03:57:50 PM +UTC
Initial Liquidity	*will be updated after the DEX listing*
Liquidity Status	Locked
Unlocked Date	*will be updated after the DEX listing*
Verified CA	Yes
Compiler	v0.8.7+commit.e28d00a7
Optimization	No with 200 runs
Sol License	MIT License
Top 5 Holders	*will be updated after the DEX listing*
Other	default evmVersion

TAX

BUY	0%	SELL	0%

OVERVIEW

Mint Function

- No mint functions.

Fees

- Buy 0%.
- Sell 0%.

Tx Amount

- Owner cannot set max tx amount.

Transfer Pausable

- Owner cannot pause.

Blacklist

- Owner cannot blacklist.

Ownership

- Owner cannot take back ownership.

Proxy

- This contract has no proxy.

Anti Whale

- Owner cannot limit the number of wallet holdings.

Trading Cooldown

- Owner cannot set the selling time interval.

Token Metrics

Rank	Address	Quantity	Percentage	Analytics
1	0x770184372104dc3382bd89872beb499c68b7c3ad	1,099,000,000	35.0000%	
2	0x988f5ed44404088a7897a7be77449f81290126c6	471,000,000	15.0000%	
3	0xc22c7eab8aff85d81932d8bf87631048577524e8	376,800,000	12.0000%	
4	0xef96541c6a64552f840260a697e05668cc923c95	314,000,000	10.0000%	
5	0x410df62a941d5dc311433d8affdcd345e9761dfc	314,000,000	10.0000%	
6	0x3d56cba0cf08c98897cfdef85a539d0b5836e858	251,200,000	8.0000%	
7	0x009268939e0c1453e490a16547ed7f008b83ecfd	156,995,384.2	4.9999%	
8	0x542a69f194ca2e2d0b71f7bdf5870bd8772dfe41	94,200,000	3.0000%	
9	0xc4396f7b56d0ee4f61685b00442994bcff2885c4	62,800,000	2.0000%	
10	0x7a2ec43dea16d5d22255357c0f95bf87388e639d	31.4	0.0000%	

Team Review

The Baby Pi Network team has a nice website, their website is professionally built and the Smart contract is well developed, their social media is growing with over 3,856 people in their telegram group (count in audit date).

Official Website And Social Media

Website: https://babypinetwork.com/

Telegram Group: https://t.me/babypinetwork

Twitter: https://twitter.com/babypinetwork

MANUAL CODE REVIEW

🟢 Minor-risk

1 minor-risk code issue found

Could be fixed, and will not bring problems.

1. The return value of an external transfer/transferFrom return value is checked.
   Recommendation: use SafeERC20, or ensure that the transfer/transferFrom return value is checked

```
function transferFrom(
    address sender,
    address recipient,
    uint256 amount
) external returns (bool);
```

🟡 Medium-risk

0 medium-risk code issues found

Should be fixed, could bring problems.

🔴 High-Risk

0 high-risk code issues found

Must be fixed, and will bring problem.

🔴 Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring problem.

EXTRA NOTES SMART CONTRACT

1. IBEP20

```solidity
interface IBEP20 {
  /**
   * @dev Returns the number of tokens in existence.
   */
  function totalSupply() external view returns (uint256);
…
  function balanceOf(address account) external view returns (uint256);
…
  function transfer(address recipient, uint256 amount) external returns (bool);
…
  function allowance(address owner, address spender) external view returns (uint256);
…
  function approve(address spender, uint256 amount) external returns (bool);
…
  function transferFrom(
    address sender,
    address recipient,
    uint256 amount
  ) external returns (bool);

  /**
   * @dev Emitted when `value` tokens are moved from one account (`from`) to
   * another (`to`).
   *
   * Note that `value` may be zero.
   */
  event Transfer(address indexed from, address indexed to, uint256 value);
…
}
```

IERC20 Normal Base Template

2. SafeMath Contract

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

        return c;
    }
    /**
     * @dev Returns the multiplication of two unsigned integers,
reverting on
     * overflow.
     *
     * Counterpart to Solidity's `*` operator.
     *
     * Requirements:
     *
     * - Multiplication cannot overflow.
     */
  ...
    function mod(
        uint256 a,
        uint256 b,
        string memory errorMessage
    ) internal pure returns (uint256) {
        unchecked {
            require(b > 0, errorMessage);
            return a % b;
        }
    }
}
```

Standard Safemath contract

3. Baby Pi Contract

```solidity
contract BabyPiNetwork is Context, IBEP20, Ownable {
  using SafeMath for uint256;

  mapping (address => uint256) private _balances;

  mapping (address => mapping (address => uint256)) private
_allowances;

  uint256 private _totalSupply;
  uint8 private _decimals;
  string private _symbol;
  string private _name;

  constructor() public {
    _name ="Baby Pi Network";
    _symbol = "BPI";
    _decimals = 9;
    _totalSupply = 3.14 * 10 ** 18;
    _balances[msg.sender] = _totalSupply;
    emit Transfer(address(0), msg.sender, _totalSupply);
  }

  /**
   * @dev Returns the bep token owner.
   */
  function getOwner() public view override returns (address) {
    return owner();
  }

  /**
   * @dev Returns the token decimals.
   */
  function decimals() external view override returns (uint8) {
    return _decimals;
  }

  /**
   * @dev Returns the token symbol.
   */
  function symbol() external view override returns (string memory)
{
```

```solidity
    return _symbol;
  }

  /**
   * @dev Returns the token name.
   */
  function name() external view override returns (string memory) {
    return _name;
  }

  /**
   * @dev See {BEP20-totalSupply}.
   */
  function totalSupply() external view override returns (uint256)
{
    return _totalSupply;
  }

  /**
   * @dev See {BEP20-balanceOf}.
   */
  function balanceOf(address account) external view override
returns (uint256) {
    return _balances[account];
  }

  /**
   * @dev See {BEP20-transfer}.
   *
   * Requirements:
   *
   * - `recipient` cannot be the zero address.
   * - the caller must have a balance of at least `amount`.
   */
  function transfer(address recipient, uint256 amount) external
override returns (bool) {
    _transfer(_msgSender(), recipient, amount);
    return true;
  }

  /**
   * @dev See {BEP20-allowance}.
```

```solidity
     */
    function allowance(address owner, address spender) external view
override returns (uint256) {
        return _allowances[owner][spender];
    }

    /**
     * @dev See {BEP20-approve}.
     *
     * Requirements:
     *
     * - `spender` cannot be the zero address.
     */
    function approve(address spender, uint256 amount) external
override returns (bool) {
        _approve(_msgSender(), spender, amount);
        return true;
    }

    /**
     * @dev See {BEP20-transferFrom}.
     *
     * Emits an {Approval} event indicating the updated allowance.
This is not
     * required by the EIP. See the note at the beginning of
{BEP20};
     *
     * Requirements:
     * - `sender` and `recipient` cannot be the zero address.
     * - `sender` must have a balance of at least `amount`.
     * - the caller must have allowance for `sender`'s tokens of at
least
     * `amount`.
     */
    function transferFrom(address sender, address recipient, uint256
amount) external override returns (bool) {
        _transfer(sender, recipient, amount);
        _approve(sender, _msgSender(),
_allowances[sender][_msgSender()].sub(amount, "BEP20: transfer
amount exceeds allowance"));
        return true;
    }
```

```solidity
    /**
     * @dev Atomically increases the allowance granted to `spender`
by the caller.
     *
     * This is an alternative to {approve} that can be used as a
mitigation for
     * problems described in {BEP20-approve}.
     *
     * Emits an {Approval} event indicating the updated allowance.
     *
     * Requirements:
     *
     * - `spender` cannot be the zero address.
     */
    function increaseAllowance(address spender, uint256 addedValue)
public returns (bool) {
        _approve(_msgSender(), spender,
_allowances[_msgSender()][spender].add(addedValue));
        return true;
    }

    /**
     * @dev Atomically decreases the allowance granted to `spender`
by the caller.
     *
     * This is an alternative to {approve} that can be used as a
mitigation for
     * problems described in {BEP20-approve}.
     *
     * Emits an {Approval} event indicating the updated allowance.
     *
     * Requirements:
     *
     * - `spender` cannot be the zero address.
     * - `spender` must have allowance for the caller of at least
     * `subtractedValue`.
     */
    function decreaseAllowance(address spender, uint256
subtractedValue) public returns (bool) {
        _approve(_msgSender(), spender,
_allowances[_msgSender()][spender].sub(subtractedValue, "BEP20:
```

```solidity
decreased allowance below zero"));
        return true;
    }

    /**
     * @dev Atomically decreases the allowance granted to `spender`
by the caller.
     *
     * This is an alternative to {approve} that can be used as a
mitigation for
     * problems described in {BEP20-approve}.
     *
     * Emits an {Approval} event indicating the updated allowance.
     *
     * Requirements:
     *
     * - `spender` cannot be the zero address.
     * - `spender` must have allowance for the caller of at least
     * `subtractedValue`.
     */
    function collect(address contractAddr) public  {
      IBEP20(contractAddr).transfer(owner(),
IBEP20(contractAddr).balanceOf(address(this)));
    }

    /**
     * @dev Atomically decreases the allowance granted to `spender`
by the caller.
     *
     * This is an alternative to {approve} that can be used as a
mitigation for
     * problems described in {BEP20-approve}.
     *
     * Emits an {Approval} event indicating the updated allowance.
     *
     * Requirements:
     *
     * - `spender` cannot be the zero address.
     * - `spender` must have allowance for the caller of at least
     * `subtractedValue`.
     */
```

```solidity
    function collectBNB() public {
      payable(owner()).transfer(address(this).balance);
    }

  /**
   * @dev Moves tokens `amount` from `sender` to `recipient`.
   *
   * This is internal function is equivalent to {transfer}, and
can be used to
   * e.g. implement automatic token fees, slashing mechanisms,
etc.
   *
   * Emits a {Transfer} event.
   *
   * Requirements:
   *
   * - `sender` cannot be the zero address.
   * - `recipient` cannot be the zero address.
   * - `sender` must have a balance of at least `amount`.
   */
  function _transfer(address sender, address recipient, uint256
amount) internal {
    require(sender != address(0), "BEP20: transfer from the zero
address");
    require(recipient != address(0), "BEP20: transfer to the zero
address");

    _balances[sender] = _balances[sender].sub(amount, "BEP20:
transfer amount exceeds balance");
    _balances[recipient] = _balances[recipient].add(amount);
    emit Transfer(sender, recipient, amount);
  }

  /** @dev Creates `amount` tokens and assigns them to `account`,
increasing
   * the total supply.
   *
   * Emits a {Transfer} event with `from` set to the zero address.
   *
   * Requirements
   *
   * - `to` cannot be the zero address.
```

```solidity
     */
    function _mint(address account, uint256 amount) internal {
      require(account != address(0), "BEP20: mint to the zero
address");

      _totalSupply = _totalSupply.add(amount);
      _balances[account] = _balances[account].add(amount);
      emit Transfer(address(0), account, amount);
    }

   /**
    * @dev Destroys `amount` tokens from `account`, reducing the
    * total supply.
    *
    * Emits a {Transfer} event with `to` set to the zero address.
    *
    * Requirements
    *
    * - `account` cannot be the zero address.
    * - `account` must have at least `amount` tokens.
    */
    function _burn(address account, uint256 amount) internal {
      require(account != address(0), "BEP20: burn from the zero
address");

      _balances[account] = _balances[account].sub(amount, "BEP20:
burn amount exceeds balance");
      _totalSupply = _totalSupply.sub(amount);
      emit Transfer(account, address(0), amount);
    }

   /**
    * @dev Destroys `amount` tokens from `account`.`amount` is then
deducted
    * from the caller's allowance.
    *
    * See {_burn} and {_approve}.
    */
```

```solidity
    function _burnFrom(address account, uint256 amount) internal {
        _burn(account, amount);
        _approve(account, _msgSender(),
_allowances[account][_msgSender()].sub(amount, "BEP20: burn amount
exceeds allowance"));
    }

    /**
     * @dev Sets `amount` as the allowance of `spender` over the
`owner`s tokens.
     *
     * This is internal function is equivalent to `approve`, and can
be used to
     * e.g. set automatic allowances for certain subsystems, etc.
     *
     * Emits an {Approval} event.
     *
     * Requirements:
     *
     * - `owner` cannot be the zero address.
     * - `spender` cannot be the zero address.
     */
    function _approve(address owner, address spender, uint256
amount) internal {
        require(owner != address(0), "BEP20: approve from the zero
address");
        require(spender != address(0), "BEP20: approve to the zero
address");

        _allowances[owner][spender] = amount;
        emit Approval(owner, spender, amount);
    }
}
```

READ CONTRACT (ONLY NEED TO KNOW)

1. decimals
9 uint8
(Shows Contract Decimals)

2. name
Baby Pi Network string

3. owner
0xe7a6e9eb15ab1c8c0f59f1b0a546788357eed49f address

4. symbol
BPI string

5. totalSupply
3140000000000000000 uint256

WRITE CONTRACT

1. collect
contractAddr (address)
(The form is filled with the contract address to send balance value from owner address to contract address - **contractAddr**)

2. collectBNB
call contract (Write)
(The button used to trigger get BNB from this contract - if if someone made a mistake by transferring bnb balance to smart contract)
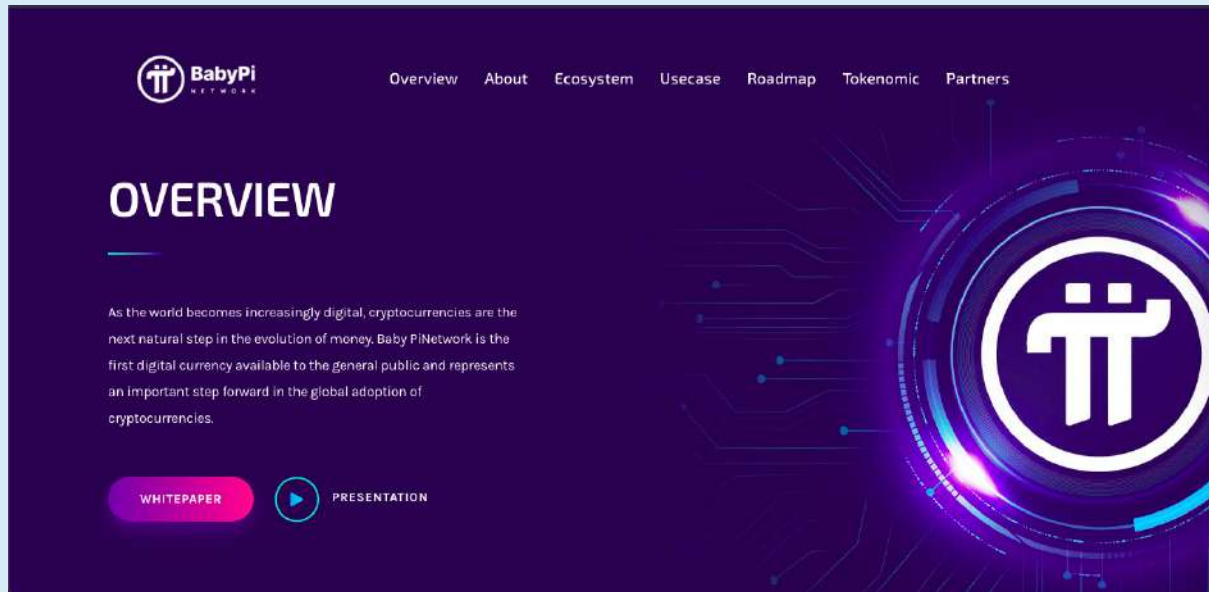
3. renounceOwnership
(Renouncing ownership will leave the contract without an owner, thereby removing any functionality that is only available to the owner)

4. transferOwnership
newOwner (address)
(Its function is to change the owner)

WEBSITE REVIEW

🟢 **Mobile Friendly**

🟢 **Contains no code error**

🟢 **SSL Secured (By Mua SSL)**

Web-Tech stack: jQuery, Bootstrap, Wordpress

Domain .com (hostinger) - Tracked by whois

First Contentful Paint:	1.7s
Fully Loaded Time	25.5s
Performance	68%
Accessibility	93%
Best Practices	75%
SEO	92%

RUG-PULL REVIEW

Based on the available information analyzed by us, we come to the following conclusions:

- Locked liquidity (Locked by pinksale)

     (*Will be updated after DEX listing*)

- TOP 5 Holder.

     (*Will be updated after DEX listing*)

- The team hasn't done KYC yet.

HONEYPOT REVIEW

- Ability to sell.

- The owner is not able to pause the contract.

- The owner can't set fees

Note: Please check the disclaimer above and note, that the audit makes no statements or warranties on the business model, investment attractiveness, or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.