

DESSERT
FINANCE



SEDUCE (SEDUCE)

Light Audit

Performed at block **16615091**

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: **0x394Dc07dA3746Bd2975055e675d6026C5756b9c4**

INITIAL DISCLAIMER

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

Dessert Finance 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.

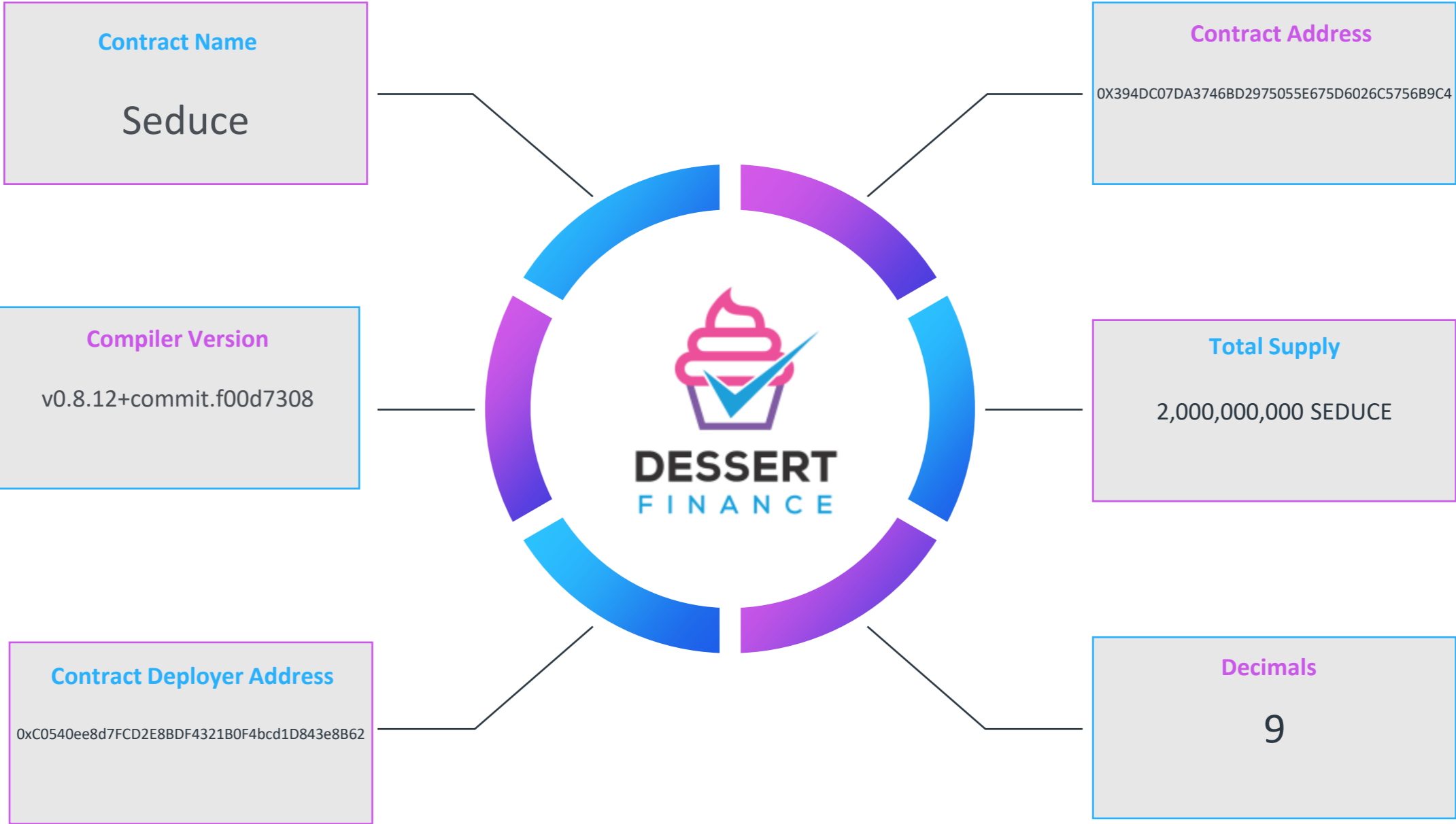
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support,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

Table of Contents



1. Contract Code Audit – Token Overview
2. BEP-20 Contract Code Audit – Overview
3. BEP-20 Contract Code Audit – Vulnerabilities Checked
4. Disclaimers

Contract Code Audit – Token Overview



BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on SEDUCE (SEDUCE)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

interface IERC20 {
    function totalSupply() external view returns (uint256);
    function decimals() external view returns (uint8);
    function symbol() external view returns (string memory);
    function name() external view returns (string memory);
    function getOwner() external view returns (address);
    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);
    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed owner, address indexed spender, uint256 value);
}

interface IFactoryV2 {
    event PairCreated(address indexed token0, address indexed token1, address lpPair, uint);
    function getPair(address token0, address token1) external view returns (address lpPair);
    function createPair(address token0, address token1) external returns (address lpPair);
}

interface IVPair {
    function factory() external view returns (address);
    function getReserves() external view returns (uint112 reserved0, uint112 reserved1, uint32 blockTimestamp);
    function get() external;
}

interface IRouter01 {
    function factory() external pure returns (address);
    function getWeth() external pure returns (address);
    function addLiquidityETH(
        address token,
        uint amountTokenDesired,
        uint amountTokenMin,
        uint amountETHMin,
        address to,
        uint deadline
    ) external payable returns (uint amountToken, uint amountETH, uint liquidity);
    function addLiquidity(
        address tokenA,
        address tokenB,
        uint amountA,
        uint amountB,
        uint amountDesired,
        uint amountMinA,
        uint amountMinB,
        address to,
        uint deadline
    ) external payable returns (uint amountA, uint amountB, uint liquidity);
    function getAmountOut(uint amountIn, address[] calldata path) external view returns (uint[] memory amounts);
    function getAmountIn(uint amountOut, address[] calldata path) external view returns (uint[] memory amounts);
}

interface IRouter02 is IRouter01 {
    function swapExactTokensForTokens(
        uint amountIn,
        uint amountOutMin,
        address[] calldata path,
        address to,
        uint deadline
    ) external payable returns (uint[] memory amounts);
}
```

Contract Address

0x394Dc07dA3746Bd2975055e675d6026C5756b9c4

TokenTracker

SEDUCE (SEDUCE)

Contract Creator

0xc0540ee8d7fcd2e8bdf4321b0f4bcd1d843e8b62

Source Code

Contract Source Code Verified

Contract Name

Seduce

Other Settings

default evmVersion, MIT

Compiler Version

v0.8.12+commit.f00d7308

Optimization Enabled

Yes with 5000 runs

Code is truncated to fit the constraints of this document.

[The code in its entirety can be viewed here.](#)

BEP-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	Scan	Result
Compiler Errors	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	✓ Low / No Risk
Integer Overflow	Complete	✓ Low / No Risk
Integer Underflow	Complete	✓ Low / No Risk
Floating Pragma	Complete	✓ Low Risk
Timestamp Dependency for Crucial Functions	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	✓ Low / No Risk
State Variable Default Visibility (x5)	Complete	✓ Low Risk

The contract code is **verified** on BSCScan.

The vulnerabilities listed above were not found in the token's Smart Contract.

Disclaimer



The opinions expressed in this document are for general informational purposes only and are **not intended to provide specific advice or recommendations for any individual or on any specific investment**. It is only intended to provide education and public knowledge regarding projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.



Thank You

DESSERT FINANCE LIGHT AUDIT HAS BEEN COMPLETED FOR SEDUCE (SEDUCE)
THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)

www.dessertswap.finance
<https://t.me/dessertswap>