



**DESSERT**  
FINANCE

## Accumulator Finance

Shimmer ERC-20 Audit  
Performed at block **1013759**

PERFORMED BY DESSERT FINANCE

FOR VAULT CONTRACT ADDRESS: `0x11C0A85EAa5b8a703267231CA5C0e32038C0ffc`

FOR STRATEGY CONTRACT ADDRESS: `0xF1869ed8218AAD228Ac4A8E8070a76C2285d89A2`

**VERIFY THIS REPORT IN THE [@DESSERTSWAP](#) TELEGRAM, CLICK HERE**

## 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.

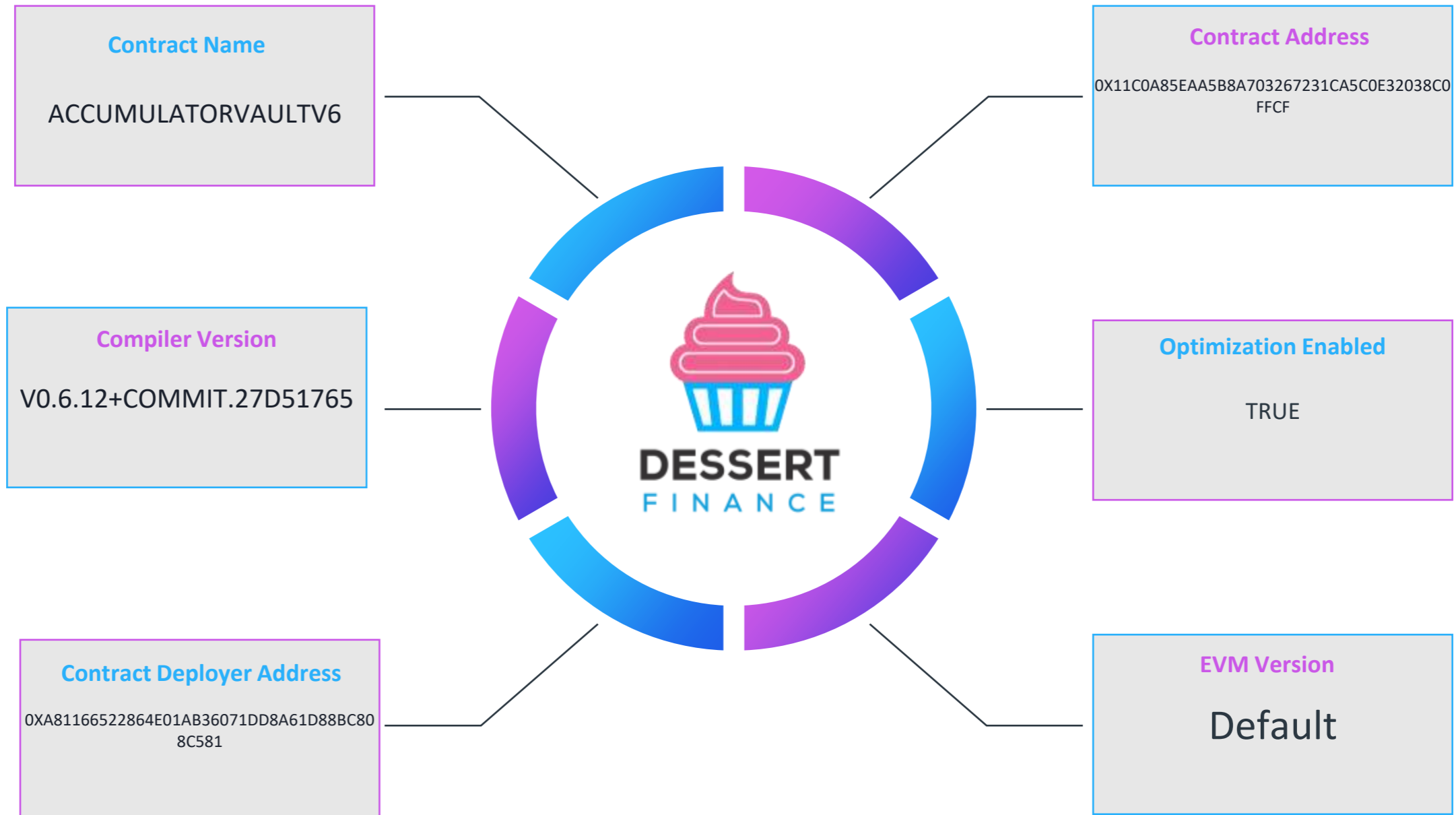
# Table of Contents



1. Contract Code Audit – Token Overview
2. ERC-20 Contract Code Audit – Overview
3. ERC-20 Contract Code Audit – Vulnerabilities Checked
4. Contract Code Audit – Contract Ownership
5. Contract Code Audit – Owner Accessible Functions
6. Liquidity Ownership – Locked / Unlocked
7. Contract Code Audit – Mint Functions
8. Contract Transaction Fees
9. Website Overview
10. Social Media
11. Top Token Holders/Wallets
12. Location Audit
13. Review of Team
14. Roadmap
15. Disclaimers

# Contract Code Audit – Token Overview

## VAULT





# ERC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on AccumulatorVaultV6

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
import "@openzeppelin/contracts/math/SafeMath.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/ReentrancyGuard.sol";

import "../interfaces/beefy/IStrategy.sol";

/**
 * @dev Implementation of a vault to deposit funds for yield optimizing.
 * This is the contract that receives funds and that users interface with.
 * The yield optimizing strategy itself is implemented in a separate 'Strategy' contract.
 */
contract AccumulatorVaultV6 is ERC20, Ownable, ReentrancyGuard {
    using SafeERC20 for IERC20;
    using SafeMath for uint256;

    struct StratCandidate {
        address implementation;
        uint proposedTime;
    }

    // The last proposed strategy to switch to.
    StratCandidate public stratCandidate;
    // The strategy currently in use by the vault.
    IStrategy public strategy;
    // The minimum time it has to pass before a strat candidate can be approved.
    uint256 public immutable approvalDelay;

    event NewStratCandidate(address implementation);
    event UpgradeStrat(address implementation);

    /**
     * @dev Sets the value of {token} to the token that the vault will
     * hold as underlying value. It initializes the vault's own 'moo' token.
     * This token is minted when someone does a deposit. It is burned in order
     * to withdraw the corresponding portion of the underlying assets.
     * @param _strategy the address of the strategy.
     * @param _name the name of the vault token.
     * @param _symbol the symbol of the vault token.
     * @param _approvalDelay the delay before a new strat can be approved.
     */
    constructor (
        IStrategy _strategy,
        string memory _name,
        string memory _symbol,
        uint256 _approvalDelay
    ) public ERC20(
        _name,
        _symbol
    ) {
```

## Contract Address

0x11C0A85EAa5b8a703267231CA5C0e32038C0ffc

## Verified At

Oct 17 2023 10:26:12 AM (-04:00 UTC)

## Contract Creator

0xa81166522864e01aB36071dd8A61D88bc808c581

## Source Code File Path

contracts/BIFI/vaults/AccumulatorVaultV6.sol

## Contract Name

AccumulatorVaultV6

## Other Settings

N/A

## Compiler Version

v0.6.12+commit.27d51765

## Optimization Enabled

true w/800 runs

Code is truncated to fit the constraints of this document.

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

# ERC-20 Contract Code Audit – Vulnerabilities Checked - VAULT

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on Shimmer.network.

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

# Contract Code Audit – Contract Ownership - VAULT

Contract Ownership has not been renounced at the time of Audit



The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

`0xa81166522864e01ab36071dd8a61d88bc808c581`

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

# Contract Code Audit – Owner Accessible Functions

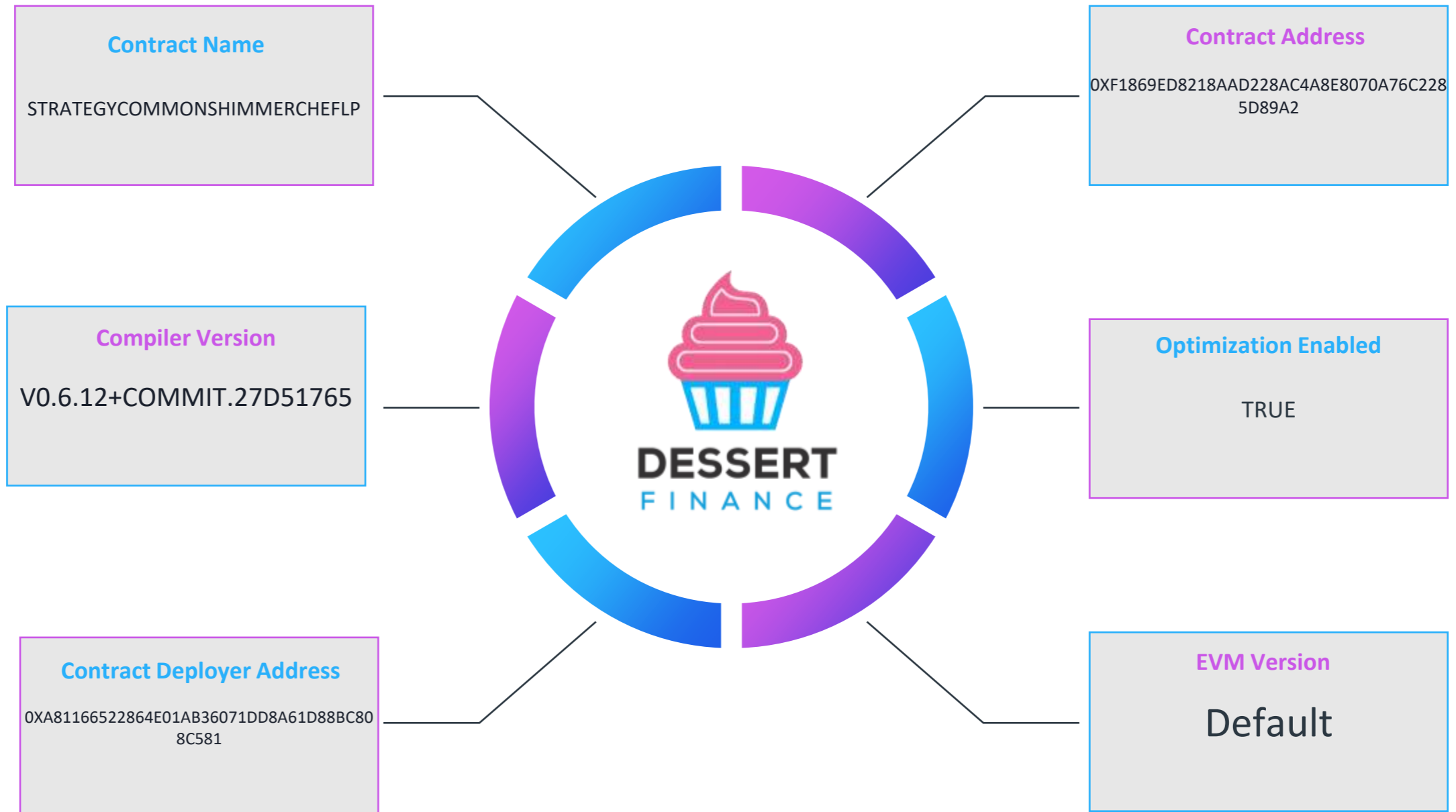
Function Name	Parameters	Visibility	Audit Notes
proposeStrat	address_implementation	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
upgradeStrat		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
inCaseTokensGetStuck	address_token	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

The functions listed above can be called by the contract owner.

If contract ownership has been renounced there is no way for the above listed functions to be called.



# Contract Code Audit – Token Overview STRATEGY



# ERC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on StrategyCommonShimmerChefLP

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
import "@openzeppelin/contracts/math/SafeMath.sol";

import "../Interfaces/common/IUniswapRouterETH.sol";
import "../Interfaces/shimmersea/ITangleseaPair.sol";
import "../Interfaces/shimmersea/IMasterChef.sol";
import "../Common/StratManager.sol";
import "../Common/FeeManager.sol";
import "../utils/StringUtils.sol";
import "../utils/GasThrottler.sol";

contract StrategyCommonShimmerChefLP is StratManager, FeeManager, GasThrot
    using SafeERC20 for IERC20;
    using SafeMath for uint256;

    // Tokens used
    address public native;
    address public output;
    address public want;
    address public lpToken0;
    address public lpToken1;

    // Third party contracts
    address public chef;
    uint256 public poolId;

    bool public harvestOnDeposit;
    uint256 public lastHarvest;
    string public pendingRewardsFunctionName;

    // Routes
    address[] public outputToNativeRoute;
    address[] public outputTolp0Route;
    address[] public outputTolp1Route;

    event StratHarvest(address indexed harvester, uint256 wantHarvested, u
    event Deposit(uint256 tvl);
    event Withdraw(uint256 tvl);
    event ChargedFees(uint256 callFees, uint256 beefyFees, uint256 strateg

    constructor(
        address _want,
        uint256 _poolId,
        address _chef,
        address _vault,
        address _unirouter,
        address _keeper,
        address _strategist,
        address _beefyFeeRecipient
```

## Contract Address

0xF1869ed8218AAD228Ac4A8E8070a76C2285d89A2

## Verified At

Oct 17 2023 10:22:57 AM (-04:00 UTC)

## Contract Creator

0xa81166522864e01aB36071dd8A61D88bc808c581

## Source Code File Path

contracts/BIFI/strategies/Common/StrategyCommonShimmerChefLP.sol

## Contract Name

StrategyCommonShimmerChefLP

## Other Settings

N/A

## Compiler Version

v0.6.12+commit.27d51765

## Optimization Enabled

true w/800 runs

Code is truncated to fit the constraints of this document.

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

The contract code is **verified** on Shimmer.network.

# ERC-20 Contract Code Audit – Vulnerabilities Checked - STRATEGY

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on Shimmer.network.

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

# Contract Code Audit – Contract Ownership - STRATEGY

Contract Ownership has not been renounced at the time of Audit



The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

`0xa81166522864e01ab36071dd8a61d88bc808c581`

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

# Contract Code Audit – Owner Accessible Functions - STRATEGY

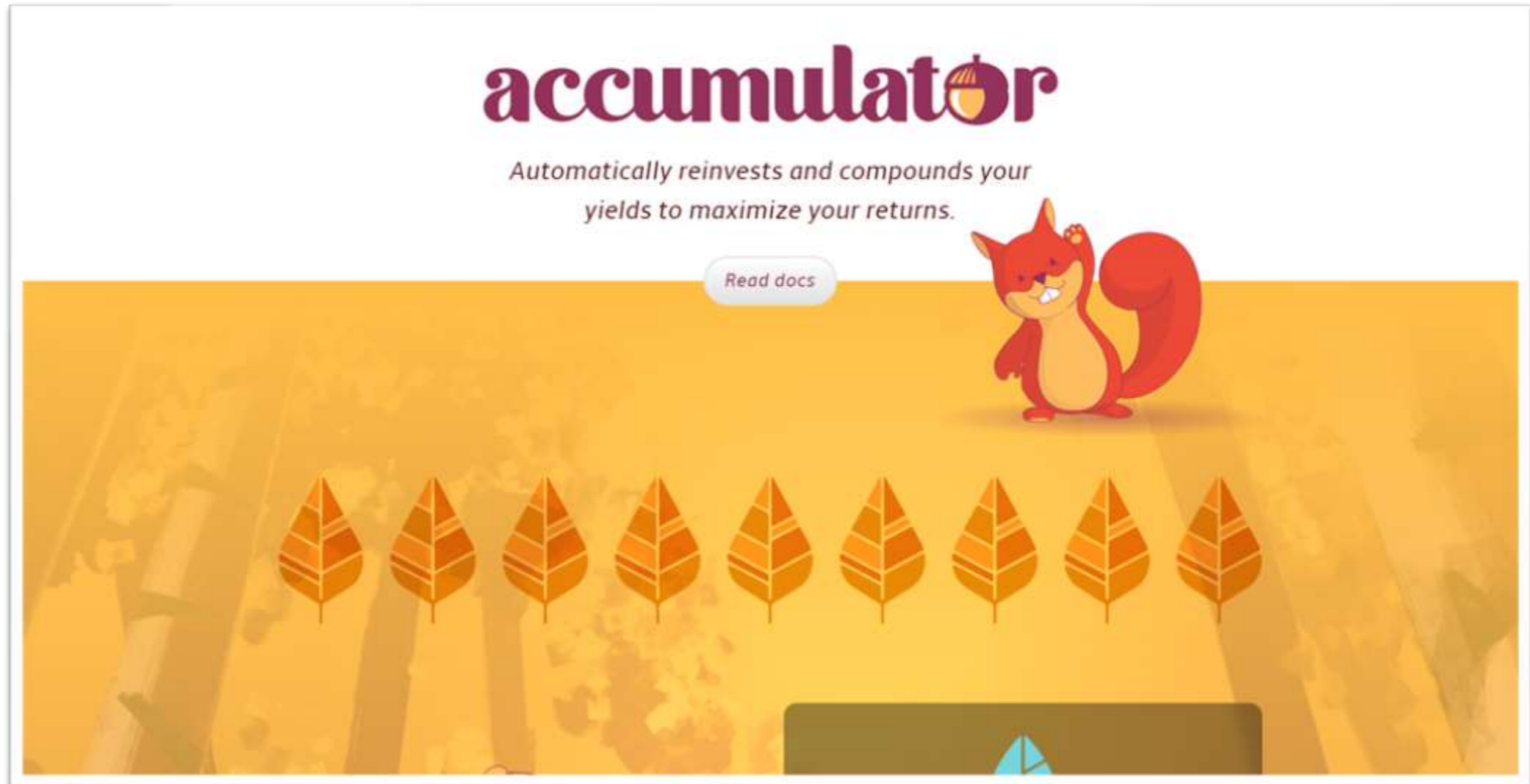
Function Name	Parameters	Visibility	Audit Notes
managerHarvest		external	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
setPendingRewardsFunctionName	string calldata _pendingRewardsFunctionName	external	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
setHarvestOnDeposit	bool _harvestOnDeposit	external	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
setShouldGasThrottle	bool _shouldGasThrottle	external	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
panic		public	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
pause		public	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.
unpause		external	onlyManager modifier is detected. Owner can call this function if the contract is not renounced.

The functions listed above can be called by the contract owner.

If contract ownership has been renounced there is no way for the above listed functions to be called.



# Website Part 1 – Overview www.accumulator.finance



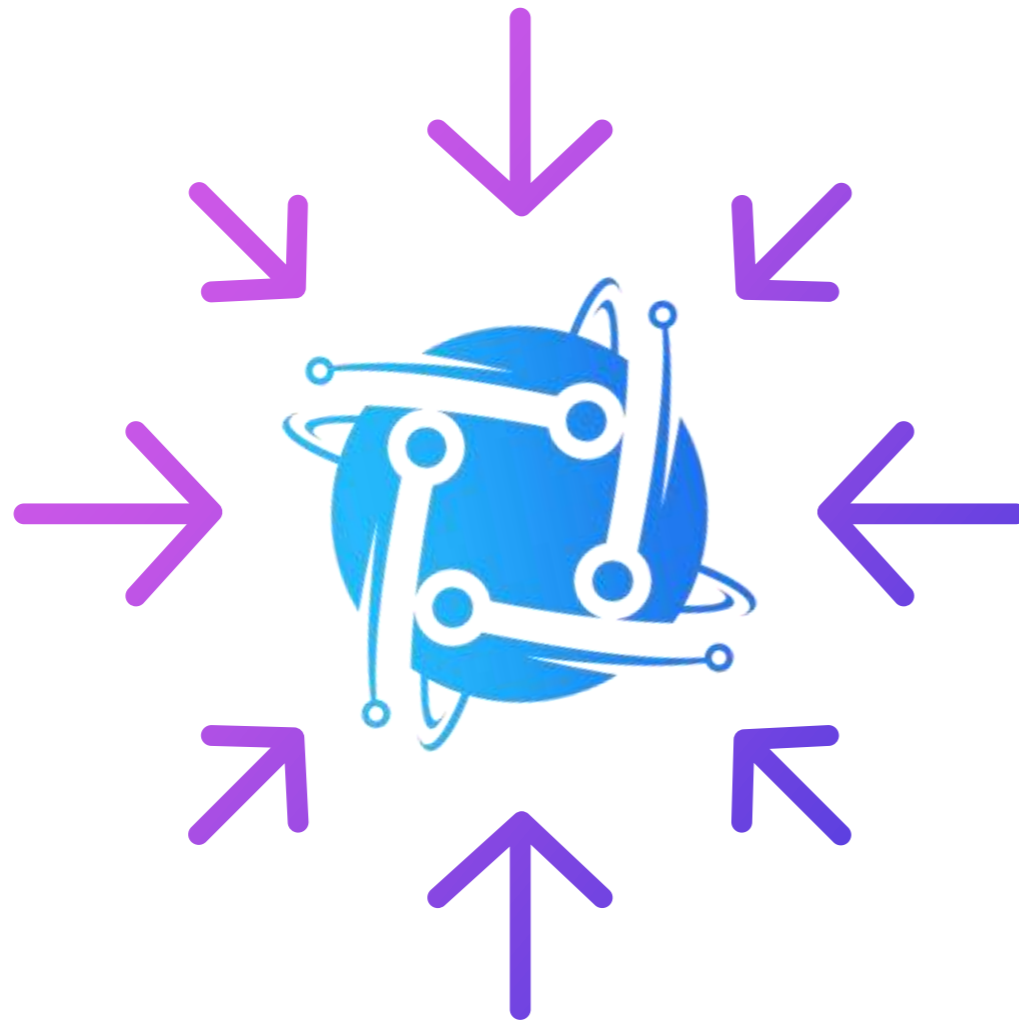
Above images are actual snapshots of the current live website of the project.

Website was registered on 04/28/2023, registration expires 04/28/2024.

**X** This meets the 3 year minimum we like to see on new projects.



## Website Part 2 – Checklist



- ✓ Mobile Friendly
- ✗ No JavaScript Errors
- ✓ Spell Check
- ✓ SSL Certificate

The website contained no severe JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https. One minor JS error was found.

No additional issues were found on the website.

# Website Part 3 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets. One minor JS error was found and shown below

```
✘ Error while trying to use the following icon www.accumulator.finance/:1  
from the Manifest: https://www.accumulator.finance/logo192.png (Download  
error or resource isn't a valid image)
```

**accumulator**

*Automatically  
reinvests and  
compounds your  
yields to maximize  
your returns.*

[Read docs](#)



# Website Part 4 (GWS) – General Web Security



## SSL CERTIFICATE

A valid SSL certificate was found. Details are as follows:

Offered to: accumulator.finance

Issued by: R3

Valid Until: Jan 2024



## CONTACT EMAIL

A valid contact email was found on the official website. Contact email is listed as shown below:

Contact

N/A



## SPAM / MALWARE / POPUPS

No malware found

No injected spam found

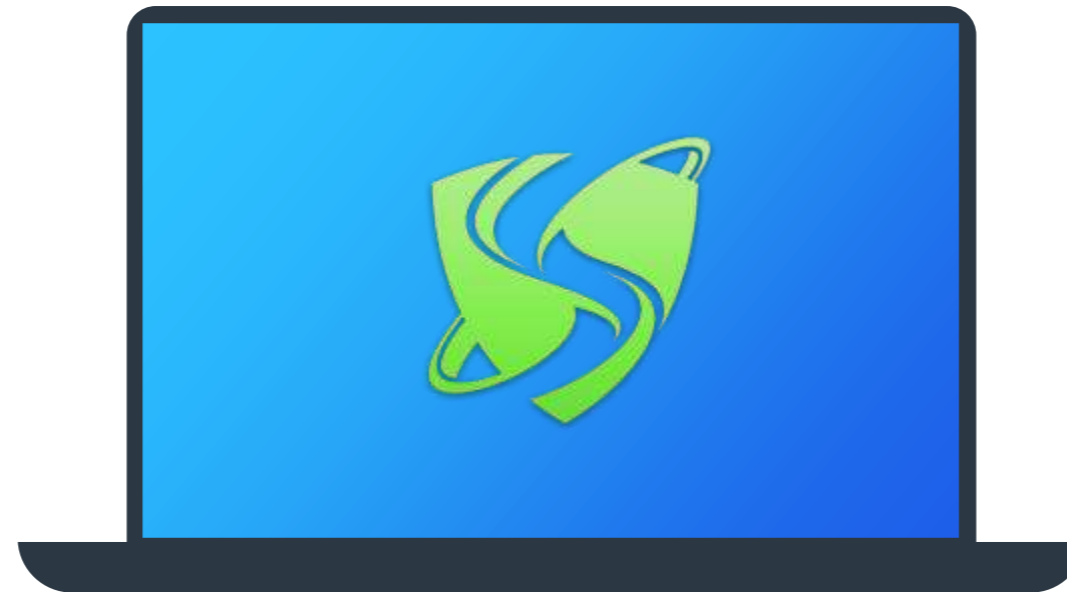
No internal server errors

No popups found

Domain is marked clean by Google, McAfee, Sucuri Labs, & ESET



# Social Media



We were able to locate a variety of Social Media networks for the project.

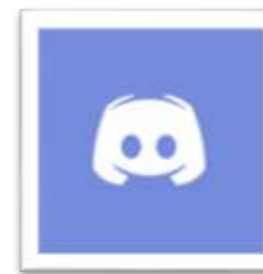
All links have been conveniently placed below.



[Twitter](#)



[Medium](#)



[Discord](#)

✓ At least 3 social media networks were found.



# Location Audit

We were unable to identify a primary location for the project at this time or a location has not been declared.



# Team Overview



We are unable to find any information about the team on the website at this time. Projects may choose to stay anonymous for a myriad of reasons.

# Roadmap

*A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.*



# 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 PROJECT AUDIT HAS BEEN COMPLETED FOR ACCUMULATOR FINANCE AT BLOCK NUMBER: **1013759**

THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)

[www.dessertswap.finance](http://www.dessertswap.finance)  
<https://t.me/dessertswap>