# YEARN V2 GENERIC-LENDER-STRAT SMART CONTRACT AUDIT

February 19, 2021

MixBytes()

# CONTENTS

# 1. INTRODUCTION

## 1.1 DISCLAIMER

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Yearn V2. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.

## 1.2 PROJECT OVERVIEW

The project is intended to provide users with tools for working with lending in DeFi. The project serves as an aggregator of all known blockchain projects for working with lending. It allows you to choose the optimal platform for the user.

# 1.3 SECURITY ASSESSMENT METHODOLOGY

At least 2 auditors are involved in the work on the audit who check the provided source code independently of each other in accordance with the methodology described below:

01 "Blind" audit includes:
> Manual code study
> "Reverse" research and study of the architecture of the code based on the source code only

Stage goal:
Building an independent view of the project's architecture
Finding logical flaws

02 Checking the code against the checklist of known vulnerabilities includes:
> Manual code check for vulnerabilities from the company's internal checklist
> The company's checklist is constantly updated based on the analysis of hacks, research and audit of the clients' code

Stage goal:
Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flashloan attacks, etc.)

03 Checking the logic, architecture of the security model for compliance with the desired model, which includes:
> Detailed study of the project documentation
> Examining contracts tests
> Examining comments in code
> Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit

Stage goal:
Detection of inconsistencies with the desired model

04 Consolidation of the reports from all auditors into one common interim report document
> Cross check: each auditor reviews the reports of the others
> Discussion of the found issues by the auditors
> Formation of a general (merged) report

Stage goal:
Re-check all the problems for relevance and correctness of the threat level
Provide the client with an interim report

05 Bug fixing & re-check.
> Client fixes or comments on every issue
> Upon completion of the bug fixing, the auditors double-check each fix and set the statuses with a link to the fix

Stage goal:
Preparation of the final code version with all the fixes

06 Preparation of the final audit report and delivery to the customer.

Findings discovered during the audit are classified as follows:

## FINDINGS SEVERITY BREAKDOWN

| Level | Description | Required action |
|---|---|---|
| Critical | Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party | Immediate action to fix issue |
| Major | Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement. | Implement fix as soon as possible |
| Warning | Bugs that can break the intended contract logic or expose it to DoS attacks | Take into consideration and implement fix in certain period |
| Comment | Other issues and recommendations reported to/acknowledged by the team | Take into consideration |

Based on the feedback received from the Customer's team regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

| Status | Description |
|---|---|
| Fixed | Recommended fixes have been made to the project code and no longer affect its security. |
| Acknowledged | The project team is aware of this finding. Recommendations for this finding are planned to be resolved in the future. This finding does not affect the overall safety of the project. |
| No issue | Finding does not affect the overall safety of the project and does not violate the logic of its work. |

# 1.4 EXECUTIVE SUMMARY

The checked volume includes a set of smart contracts that are part of the project, which combines the functionality for working with lending. The developed functionality serves as an aggregator of all known platforms for working with lending. It allows you to choose the optimal platform for the user.

# 1.5 PROJECT DASHBOARD

| | |
|---|---|
| **Client** | Yearn V2 |
| **Audit name** | Generic-lender-strat |
| **Initial version** | 979ef2f0e5da39ca59a5907c37ba2064fcd6be82<br>3ead812d7ac9844cc484a76545b3e222a9130852 |
| **Final version** | 3ead812d7ac9844cc484a76545b3e222a9130852 |
| **SLOC** | 1263 |
| **Date** | 2020-12-21 - 2021-02-19 |
| **Auditors engaged** | 2 auditors |

## FILES LISTING

| | |
|---|---|
| **Strategy.sol** | Strategy.sol |
| **AlphaHomoLender.sol** | AlphaHomoLender.sol |
| **EthCompound.sol** | EthCompound.sol |
| **EthCream.sol** | EthCream.sol |
| **GenericCompound.sol** | GenericCompound.sol |
| **GenericCream.sol** | GenericCream.sol |
| **GenericDyDx.sol** | GenericDyDx.sol |
| **GenericLenderBase.sol** | GenericLenderBase.sol |
| **IGenericLender.sol** | IGenericLender.sol |

## FINDINGS SUMMARY

| Level | Amount |
|---|---|
| **Critical** | 0 |
| **Major** | 2 |
| **Warning** | 7 |
| **Comment** | 7 |

## CONCLUSION

Smart contracts have been audited and several suspicious places have been spotted. During the audit, no critical issues were found, two issues were marked as major because it could lead to some undesired behavior, also several warnings and comments were found and discussed with the client. After working on the reported findings all of them were resolved or acknowledged (if the problem was not critical).

# 2.FINDINGS REPORT

## 2.1 CRITICAL

Not Found

## 2.2 MAJOR

| MJR-1 | It is possible to process a non-existing array element or skip an array element |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Major |
| **Status** | Fixed at 3ead812d |

### DESCRIPTION

At the line Strategy.sol#L424 is working with the elements of the `_newPositions` array in a loop.
For each element of the `lenders` array, there must be an element of the `_newPositions` array. But now the iteration of elements for the `_newPositions` array is not done correctly.
This will cause the `manualAllocation()` function to work incorrectly.

### RECOMMENDATION

It is necessary to correct the index value for the `_newPositions` array:

```
if (address(lenders[j]) == _newPositions[i].lender) {
```

| MJR-2 | Ignore failure status for `CToken` |
|---|---|
| **File** | GenericCompound.sol<br>GenericCream.sol |
| **Severity** | Major |
| **Status** | Acknowledged |

## DESCRIPTION

There are many reasons for failure `CToken`, but Lenders contracts ignore it in the all places.
Interface methods of `CToken`:
For

```
function mint(uint256 mintAmount) external returns (uint256);
```

- GenericCompound.sol#L140
- GenericCream.sol#L119

For

```
function redeemUnderlying(uint256 redeemAmount) external returns (uint256);
```

- GenericCompound.sol#L85
- GenericCompound.sol#L113
- GenericCompound.sol#L116
- GenericCream.sol#L78
- GenericCream.sol#L106
- GenericCream.sol#L109

Return value (`uint256`) is enum of errors which may be:

```
enum Error {
    NO_ERROR,
    UNAUTHORIZED,
    COMPTROLLER_MISMATCH,
    INSUFFICIENT_SHORTFALL,
    INSUFFICIENT_LIQUIDITY,
    INVALID_CLOSE_FACTOR,
    INVALID_COLLATERAL_FACTOR,
    INVALID_LIQUIDATION_INCENTIVE,
    MARKET_NOT_ENTERED, // no longer possible
    MARKET_NOT_LISTED,
    MARKET_ALREADY_LISTED,
    MATH_ERROR,
    NONZERO_BORROW_BALANCE,
    PRICE_ERROR,
    REJECTION,
    SNAPSHOT_ERROR,
    TOO_MANY_ASSETS,
    TOO_MUCH_REPAY
}
```

## RECOMMENDATION

We recommend to validate return of every method for `CToken`. If method returns no `NO_ERROR` — revert it.

## CLIENT'S COMMENTARY

> adding in some requires where useful.

## 2.3 WARNING

| WRN-1 | Safe math library not used |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Warning |
| **Status** | Fixed at 3ead812d |

### DESCRIPTION

If you do not use the library for safe math, then an arithmetic overflow may occur, which will lead to incorrect operation of smart contracts.
In the contract Strategy.sol on lines: 136, 155, 180, 206, 213, 287, 430, 464, 543, 547 calculations are without safe mathematics.

### RECOMMENDATION

All arithmetic operations need to be redone using the Safe math library.

### CLIENT'S COMMENTARY

> fixed where appropriate.

| WRN-2 | There is no processing of the value returned by the function |
|---|---|
| File | IGenericLender.sol<br>Strategy.sol |
| Severity | Warning |
| Status | Acknowledged |

## DESCRIPTION

In the IGenericLender.sol contract, line 21 has a function `withdrawAll()`. This function returns a value of type `bool`.
For the line Strategy.sol#L40, the variable `lenders` of type `IGenericLender[]` is initialized.
In the contract Strategy.sol on lines 393 and 414 there is a call to the function `withdrawAll()`.
But the return value is not processed.

## RECOMMENDATION

Add processing of the value returned by the function.

| WRN-3 | The return value is not processed when transferring tokens |
|---|---|
| **File** | GenericLenderBase.sol |
| **Severity** | Warning |
| **Status** | Fixed at **3ead812d** |

## DESCRIPTION

According to the ERC-20 specification, the `transfer()` function returns a variable of the `bool` type.
At the line GenericLenderBase.sol#L56 there is a call to the `transfer()` function. But the return value is not processed. This can lead to incorrect operation of the smart contract.

## RECOMMENDATION

It is necessary to add handling of the value returned by the `transfer()` function.

## CLIENT'S COMMENTARY

> changed to use `safeErc20`.

| WRN-4 | Gas overflow during iteration (DoS) |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Warning |
| **Status** | **No issue** |

## DESCRIPTION

Each iteration of the cycle requires a gas flow.
A moment may come when more gas is required than it is allocated to record one block. In this case, all iterations of the loop will fail.
Affected lines:
Strategy.sol#L413

## RECOMMENDATION

It is recommended adding a check for the maximum possible number of elements of the arrays.

## CLIENT'S COMMENTARY

> disagree. we don't mind this risk as manualAllocation is privileged.

| WRN-5 | Add additional check for `addLender` |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Warning |
| **Status** | **No issue** |

## DESCRIPTION

At the line Strategy.sol#L67 in method `addLender` there are no checks for `want`.

## RECOMMENDATION

It is recommended to check that `want` token of Strategy equals `want` token of Lender.

## CLIENT'S COMMENTARY

> disagree. want is checked in lender constructor.

| WRN-6 | Potential error `Index out of range` |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Warning |
| **Status** | Fixed at 3ead812d |

## DESCRIPTION

In methods:

- `estimateAdjustPosition` at the line Strategy.sol#L267 `_potential = lenders[_highest].aprAfterDeposit(toAdd)`

- `adjustPosition` at the line Strategy.sol#L399 `lenders[highest].deposit()`

- `_withdrawSome` at the line Strategy.sol#L464 `amountWithdrawn += lenders[lowest].withdraw(_amount - amountWithdrawn)`

There are risks that `lenders` array may be empty.

## RECOMMENDATION

It is recommended to add next code:

```
if (lenders.length == 0) {
    return;
}
```

| WRN-7 | Potential money remains on the strategy |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Warning |
| **Status** | Acknowledged |

## DESCRIPTION

At the line Strategy.sol#L431

`uint256 toSend = assets.mul(_newPositions[i].share).div(1000)`

Then the contract sends `toSend` amount to lender and deposits it immediately.

For example imagine that `assets` equals 33 and `lenderRatio[]` equals [{address1, 500}, {address2, 500}]. Next logic:
0. want.balanceOf(address(this)) -> 33

1. toSend = (33 * 500) // 1000 = 16 -> deposit it to address1
2. toSend = (33 * 500) // 1000 = 16 -> deposit it to address2
3. require(share == 1000, "SHARE!=1000") -> true
4. want.balanceOf(address(this)) -> 1 // remain tokens

## RECOMMENDATION

It is recommended to process remain tokens and deposit them too.

## 2.4 COMMENTS

| CMT-1 | Using magic numbers |
|---|---|
| File | Strategy.sol<br>GenericCompound.sol<br>GenericDyDx.sol<br>GenericCream.sol<br>EthCream.sol<br>EthCompound.sol<br>AlphaHomoLender.sol |
| Severity | Comment |
| Status | Fixed at 3ead812d |

### DESCRIPTION

The use in the source code of some unknown where taken values impairs its understanding.

The value is `1000`:

- in the contract Strategy.sol on lines 45, 431, 436

The value is `1e18`:

- in the contract Strategy.sol#L543
- in the contract GenericCompound.sol#L62
- in the contract GenericDyDx.sol on lines 177, 178
- in the contract GenericCream.sol#L55
- in the contract EthCream.sol#L53
- in the contract EthCompound.sol on lines 59, 181, 189, 191

The value is `1`:

- in the contract AlphaHomoLender.sol#L137
- in the contract EthCompound.sol#L108
- in the contract EthCream.sol#L102
- in the contract GenericCompound.sol#L108
- in the contract GenericCream.sol#L101
- in the contract GenericDyDx.sol#L106

### RECOMMENDATION

It is recommended that you create constants with meaningful names for using numeric values.

### CLIENT'S COMMENTARY

> explained magic numbers where appropriate. Changed in tendTrigger.

| CMT-2 | Function without logic |
|---|---|
| **File** | IGeneric.sol<br>AlphaHomoLender.sol<br>EthCompound.sol<br>EthCream.sol<br>GenericCompound.sol<br>GenericCream.sol<br>GenericDyDx.sol |
| **Severity** | Comment |
| **Status** | Acknowledged |

## DESCRIPTION

At the line IGeneric.sol#L23 has an external function `enabled()`.
This function always returns true when executed. There is no other logic in this function.
This function is located in the following locations:

- at the line AlphaHomoLender.sol#L177
- at the line EthCompound.sol#L164
- at the line EthCream.sol#L143
- at the line GenericCompound.sol#L150
- at the line GenericCream.sol#L129
- at the line GenericDyDx.sol#L160

## RECOMMENDATION

It is recommended that you remove this function or add logic to the body of the function.

| CMT-3 | The unchangeable value of the variable |
|---|---|
| **File** | Strategy.sol |
| **Severity** | Comment |
| **Status** | Acknowledged |

## DESCRIPTION

The contract Strategy.sol#L477 has an internal function `liquidatePosition()`.
One of the return variables for this function is called `_loss`.
The value of this variable is always `0`.
This can be seen on lines 482, 486, 488.

## RECOMMENDATION

It is recommended to delete a variable whose value does not change.

| CMT-4 | Maximum value in function `approve()` |
|---|---|
| **File** | GenericCream.sol<br>GenericLenderBase.sol<br>GenericCompound.sol<br>GenericDyDx.sol |
| **Severity** | Comment |
| **Status** | Acknowledged |

## DESCRIPTION

Setting a maximum value for the amount of tokens that can be manipulated after calling the `approve()` function could cause an attacker to invoke his transaction for his profit.
Such calls are now in the following places:
GenericCream.sol#L39
GenericLenderBase.sol#L45
GenericCompound.sol#L46
GenericDyDx.sol#L34

## RECOMMENDATION

When calling the `approve()` function, set the actual value for the amount of tokens.

| CMT-5 | Unresolved `TODO` |
|---|---|
| **File** | GenericDyDx.sol |
| **Severity** | Comment |
| **Status** | **Fixed** at **3ead812d** |

## DESCRIPTION

Unresolved `TODO` was found in GenericDyDx.sol#L29.

## RECOMMENDATION

It is recommended to resolve it.

| CMT-6 | Add modifier for `emergencyExit` state |
|---|---|
| File | Strategy.sol |
| Severity | Comment |
| Status | Acknowledged |

## DESCRIPTION

Some functions of Strategy.sol don't check `emergencyExit` state. This allows to continue working with the contract after exit.

## RECOMMENDATION

It is recommended fixing it with special modifier `emergencyExit`.

|  |  |
|---|---|
| CMT-7 | Add event for `migrate` |
| File | Strategy.sol |
| Severity | Comment |
| Status | Acknowledged |

## DESCRIPTION

At the line Strategy.sol#L554 for the migration process there is only a `Transfer` event.

## RECOMMENDATION

It is recommended to emit special event `Migrated` in order to keep users up to date.

## CLIENT'S COMMENTARY

> This is a base strategy improvement suggestion, out of scope.

# 3.ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

## BLOCKCHAINS

Ethereum

Cosmos

EOS

Substrate

## TECH STACK

Python

Solidity

Rust

C++

## CONTACTS

https://github.com/mixbytes/audits_public

https://mixbytes.io/

hello@mixbytes.io

https://t.me/MixBytes

https://twitter.com/mixbytes