Xend Finance: Esusu & Yearn Savings

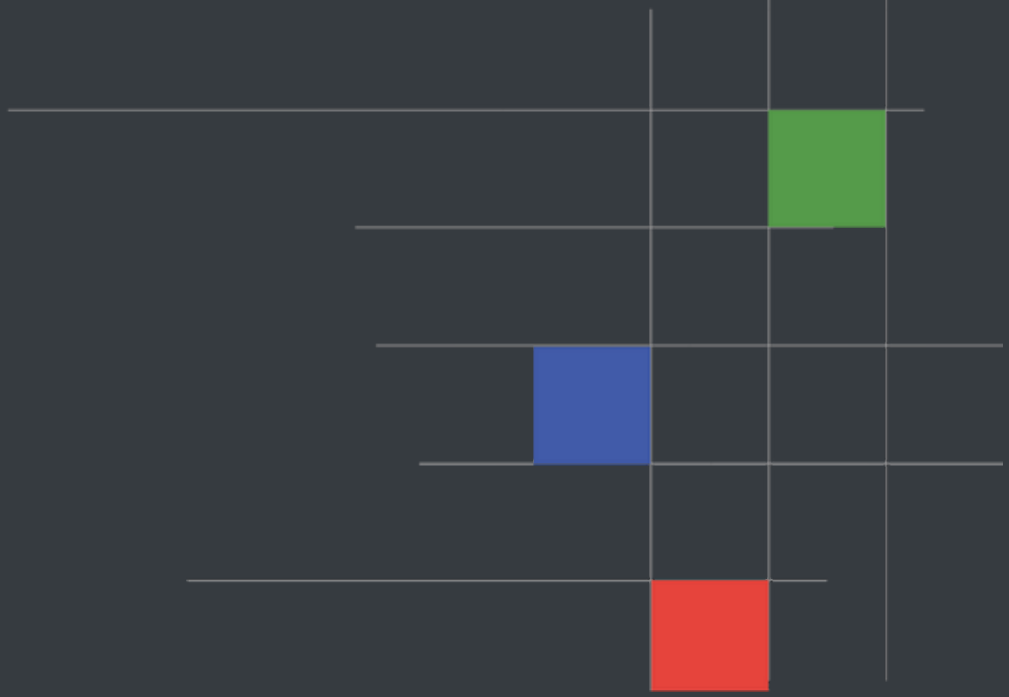
Smart Contracts

Security Assessment

February 13th, 2021

By:
Sheraz Arshad @ CertiK
sheraz.arshad@certik.org

Camden Smallwood @ CertiK
camden.smallwood@certik.org



Disclaimer

CertiK reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has indeed completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

Overview

Project Summary

Project Name	Xend Finance: Esusu & Yearn Savings
Description	The code in audit comprise of delta related to rewarding group's creator with a percentage of the commission fee and to track total $xend token rewards. Contracts, along with the lines of code audited in the delta: EsusuAdapterWithdrawalDelegate.sol L405 - L417, L539 XendFinanceGroup_Yearn_V1.sol L1089 - L1104, L1190 XendFinanceIndividual_Yearn_V1.sol, L719
Platform	Ethereum; Solidity, Yul
Codebase	Yearn Savings Esusu
Commits	1. df9d2971b0be629caa58cb410e766ea98cf1aac1 2. 0032da9c4944a1c835eded5c2e600763b41cb931 3. 1efcbe816e6eb37d193a391dc2e9d965fdea365e

Audit Summary

Delivery Date	February 13th, 2021
Method of Audit	Static Analysis, Manual Review
Consultants Engaged	2
Timeline	February 12th, 2021 - February 13th, 2021

Vulnerability Summary

Total Issues	1
Total Critical	0
Total Major	1
Total Medium	0
Total Minor	0
Total Informational	0

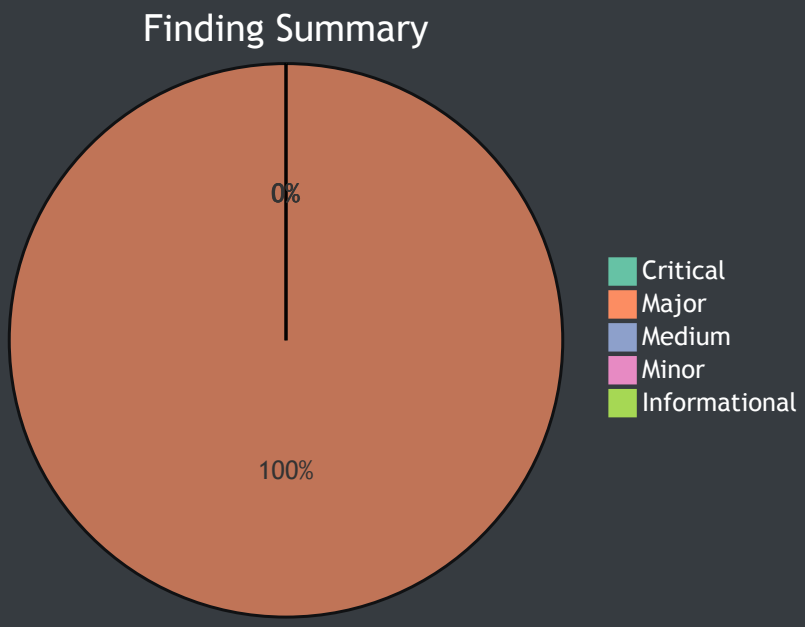
Executive Summary

This report represents the results of CertiK's engagement with Xend on the delta related to rewarding group creator with a percentage of the commission fee and to track total $xend token rewards. Only one issue was identified, XFG-01, outlining that the Group's creator reward is not deducted from the member's underlying balance.

Files In Scope

ID	Contract	Location
EAW	EsusuAdapterWithdrawalDelegate.sol	EsusuAdapterWithdrawalDelegate.sol
XFG	XendFinanceGroup_Yearn_V1.sol	XendFinanceGroup_Yearn_V1.sol
XFI	XendFinanceIndividual_Yearn_V1.sol	XendFinanceIndividual_Yearn_V1.sol

Findings



ID	Title	Type	Severity	Resolved
XFG-01	Group's creator reward is not deducted from the member's underlying balance	Volatile Code	Major	✓

XFG-01: Group's creator reward is not deducted from the member's underlying balance

Type	Severity	Location
Volatile Code	Major	XendFinanceGroup_Yearn_V1.sol L1091-L1094

Description:

The `L1093` subtracts the total fee in underlying from member's underlying balance. This deduction of fee does not take into account the fee set aside for group's creator as it is already subtracted from the total fee on `L1091`. This will potentially result in discrepancy of the underlying balance of contract as the portion of fee paid to group's creator is not subtracted from the member's underlying balance.

Recommendation:

We recommend to subtract the group creator's fee from the underlying balance of the member.

```
underlyingAmountThatMemberDepositIsWorth = underlyingAmountThatMemberDepositIsWorth
            .sub(finalAmountToChargeAsFees.add(creatorReward));
```

Alleviation:

Alleviations were applied by subtracting group's creator's reward amount from the underlying balance of the member as of commit with hash `1efcbe816e6eb37d193a391dc2e9d965fdea365e`.

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings refer to exhibits that do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation exhibits entail findings that relate to mishandling of math formulas, such as overflows, incorrect operations etc.

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Data Flow

Data Flow findings describe faults in the way data is handled at rest and in memory, such as the result of a `struct` assignment operation affecting an in-memory `struct` rather than an in-storage one.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of `private` or `delete`.

Coding Style

Coding Style findings usually do not affect the generated byte-code and comment on how to make the codebase more legible and as a result easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a `constructor` assignment imposing different `require` statements on the input variables than a setter function.

Magic Numbers

Magic Number findings refer to numeric literals that are expressed in the codebase in their raw format and should otherwise be specified as `constant` contract variables aiding in their legibility and maintainability.

Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

Dead Code

Code that otherwise does not affect the functionality of the codebase and can be safely omitted.