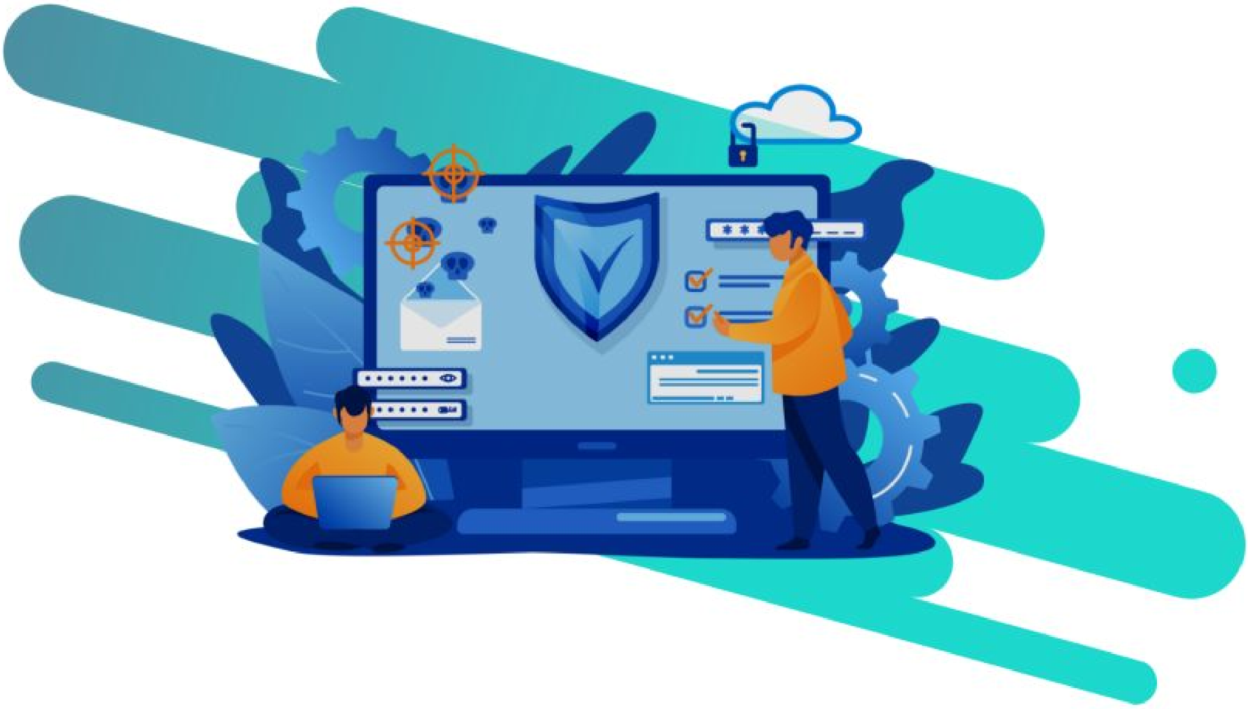
Jorge Rodriguez

# CYBERSEC CONTRACT AUDIT REPORT

## xETH-G - SECURITY REVIEW



## Introduction

During December of 2020, xETH-G team engaged CTDSec to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. xETH-G provided CTDSec with access to their code repository and whitepaper.

# Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bugfree status. The audit documentation is for discussion purposes only.

I always recommend having a bug bounty program opened to detect future bugs.

## Coverage

### Target Code and Revision

For this audit, we performed research, investigation, and review of the xETH-G contract followed by issue reporting, along with mitigation and remediation instructions outlined in this report. The following code files are considered in-scope for the review:

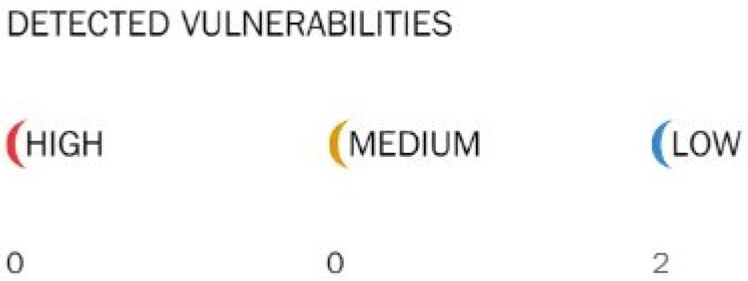
xETH-G.Sol , xETH-G-Rebaser.sol, xETH-G-safemath.sol

## Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

- Correctness of the protocol implementation [Result OK]
- User funds are secure on the blockchain and cannot be transferred without user permission [Result OK]
- Vulnerabilities within each component as well as secure interaction between the network components [Result OK]
- Correctly passing requests to the network core [Result OK]
- Data privacy, data leaking, and information integrity [Result OK]
- Susceptible to reentrancy attack [Result OK]
- Key management implementation: secure private key storage and proper management of encryption and signing keys [Result OK]
- Handling large volumes of network traffic [Result OK]
- Resistance to DDoS and similar attacks [Result OK]
- Aligning incentives with the rest of the network [Result OK]
- Any attack that impacts funds, such as draining or manipulating of funds [Result OK]
- Mismanagement of funds via transactions [Result OK]
- Inappropriate permissions and excess authority [Result OK]
- Special token issuance model [Result OK]

## DETECTED VULNERABILITIES



DETECTED VULNERABILITIES

| HIGH | MEDIUM | LOW |
|---|---|---|
| 0 | 0 | 2 |

## ARCHITECTURE

Since the image is too large we attach a link to be able to correctly view the content.

https://ibb.co/TrkdgZk

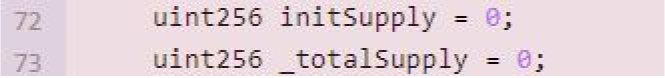
## ISSUES

### LOW SWC-000

Implicit visibility level.

The default function visibility level in contracts is public, in interfaces - external, state variable default visibility level is internal. In contracts, the fallback function can be external or public. In interfaces, all the functions should be declared as external. Explicitly define function visibility to prevent confusion.

Locations - xETH-G.Sol



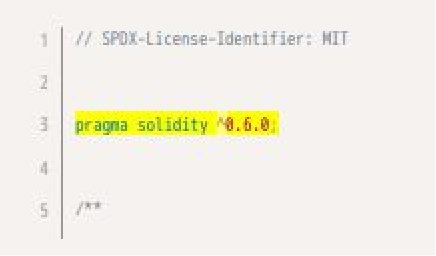
```
    uint256 initSupply = 0;
    uint256 _totalSupply = 0;
```

### LOW - SWC-103

A floating pragma is set.

The current pragma Solidity directive is "">=0.6.2"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Locations - xETH-G-safemath.sol:



```
// SPDX-License-Identifier: MIT

pragma solidity ^0.6.0;

/**
```

## Summary of the Audit

Contract is safe to deploy and good cybersecurity practices have been followed in its development.