

# antier

Decentralizing the World



## FORTIFIED X SMART CONTRACT

AUDIT REPORT - MARCH 2024

# TABLE OF CONTENTS



<b>Summary</b> .....	<b>03</b>
<b>Overview</b> .....	<b>04</b>
Project Overview .....	04
Audit Details .....	04
Vulnerability Summary .....	05
Vulnerabilities Found .....	05
<b>Vulnerable Proxy</b> .....	<b>05</b>
<b>Optimizations</b> .....	<b>06</b>
Gas Inefficient Accessibility .....	06
Un-Necessary Modifier .....	07
Centralisation Risk .....	07
<b>Technical Analysis</b> .....	<b>09</b>
<b>Limitations On Disclosure Ans Use Of This Report</b> .....	<b>10</b>

# Summary

This is a limited audit report based on our analysis of the Fortified X Smart Contract. It covers industry best practices as of the date of this report, concerning: smart contract best coding practices, cybersecurity vulnerabilities, issues in the framework, and algorithms based on white paper and code, the details of which are set out in this report, (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

Smart contracts are deployed and executed on the blockchain. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks.

You are advised to read the full report to get a full view of our analysis. While we did our best in producing this report, it is important to note that you should not rely on this report, and cannot claim against us, based on what it says or does not say, or how we produced it, and you need to conduct your independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy all copies of this report downloaded and/or printed by you.

The report is provided "as it is" without any condition, warranty, or other terms of any kind except as set out in this disclaimer. Team Antier Solutions hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose, and the use of reasonable care and skill) which, but for this clause, might affect the report.

Fortified X commissioned Antier Solutions to perform an end-to-end source code review of their Solidity Smart Contract. Team Antier Solutions performed the audit on 28th March 2024.

The following report discusses severity issues and their scope of rectification through change recommendations. It also highlights activities that are successfully executed and others that need total reworking (if any).

The report emphasizes best practices in coding and the security vulnerabilities if any.

The information in this report should be used to understand the overall code quality, security, and correctness of the Smart Contract. The analysis is static and entirely limited to the Smart Contract code.

In the audit, we reviewed the Smart Contract's code that implements Fortified X.

# Overview

## Project Overview

Project Name	Fortified X
Status	Pre Deployment
Language	Solidity
Code Repo	

## Audit Details

Audit Date	28-March-2024
Tools Used	Slither & Mythril
Audit Type	Rectification Audit
Initial Commit	4193c23cb9362031727905abdcea673c76b670e9


## Fix Commits

Commit	Commit Hash
1st	e7c594454b95be847106ea180383b75b0b5000f5

## Audit Scope

File	Status
Fortified X.sol	Safe

## Vulnerability Summary

 Decentralizing the World	High Difficulty	Medium Difficulty	Low Difficulty	Not Exploitable
Highly Vulnerable	0	0	0	0
Medium Vulnerability	0	0	0	0
Low Vulnerability	0	0	0	0
Optimization Errors	0	0	0	1

## Vulnerabilities found

Sr. No.	Vulnerability	Severity	Status
1	Vulnerable Proxy	Low	Rectified

## Vulnerable Proxy

### Vulnerability Details

Severity	Difficulty	Location	Status
Low	Low	Fortified.sol	Rectified

### Exploitable Syntax

```
1 function initialize() public initializer {
2   __ERC20_init("FXAX", "FXAS");
3   __Ownable_init();
4   maxSupply = 1e9 * 10 ** decimals();
5
6   _mint(msg.sender, maxSupply);
7 }
```

## Description

An uninitialized implementation contract can be potentially taken over by an attacker, which may impact the proxy.

## Recommendation

invoke the `_disableInitializers` function in the constructor to automatically lock it when it is deployed.

## Optimizations

Sr. No.	Title	Severity	Status
1	Gas Inefficient Accessibility	Optimization	Rectified
2	Un-Necessary Modifier	Recommendation	Rectified
3	Centralisation Risk	Recommendation	Pending

## Gas Inefficient Accessibility

### Vulnerability Details

Severity	Difficulty	Location	Status
Optimisation	Not Exploitable	Fortified.sol	Rectified

### Syntax

```
1 function burn(uint _amount) public onlyOwner {
2   require(_amount > 0, "Amount must be greater than 0");
3
4   _burn(msg.sender, _amount);
5 }
```

## Description

The function `burn` is marked `public` despite never being called internally.

## Recommendation

The function can be marked `external` for more gas efficiency.

# Un-Necessary Modifier

## Vulnerability Details

Severity	Difficulty	Location	Status
Recommendation	Not Exploitable	Fortified.sol	Rectified

## Syntax

```
1 function burn(uint _amount) public onlyOwner {  
2   require(_amount > 0, "Amount must be greater than 0");  
3  
4   _burn(msg.sender, _amount);  
5 }
```

## Suggestion

The onlyOwner modifier can be deemed unnecessary for the burn function based on the business logic since function caller can only burn his own holdings. If users are to be abrogated from burning altogether, the msg.sender state variable can be replaced with owner's address to save gas.

# Centralisation Risk

## Vulnerability Details

Severity	Difficulty	Location	Status
Recommendation	Not Exploitable	Fortified.sol	Pending

## Syntax

```
1 function initialize() public initializer {  
2   __ERC20_init("FXAX", "FXAS");  
3   __Ownable_init();  
4   maxSupply = 1e9 * 10 ** decimals();  
5  
6   _mint(msg.sender, maxSupply);  
7 }
```

### Description

The total supply for the token are minted to a single address which can be fatal i case the key is compromised/stolen. It also might affect user's trust on the token.

### Recommendation

It is recommended to store the funds in a multisig / Multi party computation wallet or a vesting contract.



# Technical Analysis

We checked Fortified X Sales Smart Contracts for commonly known and specific business logic vulnerabilities. Following is the list of vulnerabilities tested in the Smart Contract code:

Vulnerability	Results	Countermeasure Used
Reentrancy	Pass	N/A
Timestamp Dependence	N/A	-
Race Condition	Pass	-
Use Of TX. Origin	N/A	N/A
Gasless Send	N/A	N/A
Balance Equality	Pass	-
Nested Array	Pass	N/A
Unchecked External Call	Pass	-
Mathematical Errors	Pass	-
Private Modifier	Pass	-
Locked Money	Pass	-
Integer Overflow/Underflow	Pass	Solidity Version 0.8.0+
Address Hardcoded	Pass	-
Implicit Visibility Level	Pass	-

# Limitations on Disclosure and Use of this Report

This report contains information concerning potential details of Fortified X and methods for exploiting them. Antier Solutions recommends that precautions should be taken to protect the confidentiality of this document and the information contained herein.

Security Assessment is an uncertain process based on experiences, currently available information, and known threats. All information security systems, which by their nature are dependent on human beings, are vulnerable to some degree. Therefore, although Antier Solutions has identified major security vulnerabilities in the analyzed system, there can be no assurance that any exercise of this nature will identify all possible vulnerabilities or propose exhaustive and operationally viable recommendations to mitigate those exposures.

As technologies and risks change over time, the vulnerabilities associated with the operation of the Fortified X Smart Contract described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities will also change. Antier Solutions makes no undertaking to supplement or update this report based on the changed circumstances or facts of which Antier Solutions becomes aware after the date hereof.

This report may recommend that Antier Solutions use certain software or hardware products manufactured or maintained by other vendors. Antier Solutions bases these recommendations on its prior experience with the capabilities of those products. Nonetheless, Antier Solutions does not and cannot warrant that a particular product will work as advertised by the vendor, nor that it will operate in the manner intended.

The Non-Disclosure Agreement (NDA) in effect between Antier Solutions and Fortified Assets Xchange SRL Ltd. governs the disclosure of this report to all other parties, including product vendors and suppliers.