

SALUS SECURITY

MAY 2025



# CODE SECURITY ASSESSMENT

EPT CROSS

# Overview

## Project Summary

- Name: EPT - Cross
- Platform: EVM-compatible chains
- Language: Solidity
- Audit Range: See [Appendix - 1](#)

# Project Dashboard

## Application Summary

Name	EPT - Cross
Version	v3
Type	Solidity
Dates	May 22 2025
Logs	Apr 18 2025; Apr 18 2025; May 22 2025

## Vulnerability Summary

Total High-Severity issues	0
Total Medium-Severity issues	0
Total Low-Severity issues	1
Total informational issues	1
Total	2

## Contact

E-mail: [support@salusec.io](mailto:support@salusec.io)

## Risk Level Description

<b>High Risk</b>	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for clients' reputations or serious financial implications for clients and users.
<b>Medium Risk</b>	The issue puts a subset of users' sensitive information at risk, would be detrimental to the client's reputation if exploited, or is reasonably likely to lead to a moderate financial impact.
<b>Low Risk</b>	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
<b>Informational</b>	The issue does not pose an immediate risk, but is relevant to security best practices or defense in depth.

# Content

<b>Introduction</b>	<b>4</b>
1.1 About SALUS	4
1.2 Audit Breakdown	4
1.3 Disclaimer	4
<b>Findings</b>	<b>5</b>
2.1 Summary of Findings	5
2.2 Notable Findings	6
1. Centralization risk with initial token distribution	6
2.3 Informational Findings	7
2. Missing two-step transfer ownership pattern	7
<b>Appendix</b>	<b>8</b>
Appendix 1 - Files in Scope	8

# Introduction

## 1.1 About SALUS

At Salus Security, we are in the business of trust.

We are dedicated to tackling the toughest security challenges facing the industry today. By building foundational trust in technology and infrastructure through security, we help clients to lead their respective industries and unlock their full Web3 potential.

Our team of security experts employ industry-leading proof-of-concept (PoC) methodology for demonstrating smart contract vulnerabilities, coupled with advanced red teaming capabilities and a stereoscopic vulnerability detection service, to deliver comprehensive security assessments that allow clients to stay ahead of the curve.

In addition to smart contract audits and red teaming, our Rapid Detection Service for smart contracts aims to make security accessible to all. This high calibre, yet cost-efficient, security tool has been designed to support a wide range of business needs including investment due diligence, security and code quality assessments, and code optimisation.

We are reachable on Telegram (<https://t.me/salusec>), Twitter ([https://twitter.com/salus\\_sec](https://twitter.com/salus_sec)), or Email ([support@salusec.io](mailto:support@salusec.io)).

## 1.2 Audit Breakdown

The objective was to evaluate the repository for security-related issues, code quality, and adherence to specifications and best practices. Possible issues we looked for included (but are not limited to):

- Risky external calls
- Integer overflow/underflow
- Transaction-ordering dependence
- Timestamp dependence
- Access control
- Call stack limits and mishandled exceptions
- Number rounding errors
- Centralization of power
- Logical oversights and denial of service
- Business logic specification
- Code clones, functionality duplication

## 1.3 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release and does not give any warranties on finding all possible security issues with the given smart contract(s) or blockchain software, i.e., the evaluation result does not guarantee the nonexistence of any further findings of security issues.

# Findings

## 2.1 Summary of Findings

ID	Title	Severity	Category	Status
1	<a href="#">Centralization risk with initial token distribution</a>	Low	Centralization	Resolved
2	Missing two-step transfer ownership pattern	Informational	Business logic	Mitigated

## 2.2 Notable Findings

Significant flaws that impact system confidentiality, integrity, or availability are listed below.

1. Centralization risk with initial token distribution	
Severity: Low	Category: Centralization
Target: <ul style="list-style-type: none"><li>- contracts/EPTCross_main.sol</li></ul>	

### Description

contracts/EPTCross\_main.sol:L10 - L18

```
constructor(  
    string memory _name,  
    string memory _symbol,  
    address _lzEndpoint,  
    address _delegate,  
    address _treasury  
) OFT(_name, _symbol, _lzEndpoint, _delegate) Ownable(_delegate) {  
    mint(_treasury, MAX_SUPPLY);  
}
```

When the contract is deployed, \$EPT is sent to one account. This account then has full control over the token distribution. If it is an EOA account, any compromise of its private key could drastically affect the project – for example, attackers could manipulate the price of \$EPT on the DEX if they gain access to the private key.

### Recommendation

It is recommended to transfer tokens to a multi-sig account and promote transparency by providing a breakdown of the intended initial token distribution in a public location.

### Status

The team has moved privileged addresses to multi-sig addresses to mitigate this issue.

## 2.3 Informational Findings

### 2. Missing two-step transfer ownership pattern

Severity: Informational

Category: Business logic

Target:

- contracts/EPTCross\_main.sol

### Description

The `EPTCrossMain` contract inherits from the Ownable contract. This contract does not implement a two-step process for transferring ownership. Thus, ownership of the contract can easily be lost when making a mistake in transferring ownership.

### Recommendation

Consider using the [Ownable2Step](#) contract from OpenZeppelin instead.

### Status

This issue has been mitigated by the team. The team stated that due to cross-chain business requirements, they cannot fully renounce the privilege. However, the privileged address has been transferred to a multisig account, which mitigates the issue.



# Appendix

## Appendix 1 - Files in Scope

This audit covered the following files from address

<0x3Dc8e2d80b6215a1BCcAe4d38715C3520581E77c> :

File	SHA-1 hash
contracts/EPTCross_main.sol	ede84d14107c5daf5cac06305e8626a80b2d68e7