



Cyberscope

A *TAC Security* Company

Audit Report

kops

October 2025

Repository

<https://github.com/KOPs-ai/strategy.contracts/blob/main/src/MoxYieldUSDT.sol>

Commit b7d4fdfe370a7822127bf66c35918bb783b774ff

Audited by © cyberscope

Table of Contents

Table of Contents	1
Risk Classification	2
Review	3
Audit Updates	3
Source Files	3
Overview	4
Findings Breakdown	5
Diagnostics	6
CCR - Contract Centralization Risk	7
Description	7
Recommendation	7
Team Update	7
UTPD - Unverified Third Party Dependencies	8
Description	8
Recommendation	8
Team Update	8
Functions Analysis	9
Inheritance Graph	10
Flow Graph	11
Summary	12
Disclaimer	13
About Cyberscope	14

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation:** This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation:** This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical:** Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium:** Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor:** Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative:** Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
● Critical	Highly Likely / High Impact
● Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
● Minor / Informative	Unlikely / Low to no Impact

Review

Repository	https://github.com/KOPs-ai/strategy.contracts/blob/main/src/MaxYieldUSDT.sol
Commit	b7d4fdfe370a7822127bf66c35918bb783b774ff

Audit Updates

Initial Audit	07 Oct 2025
----------------------	-------------

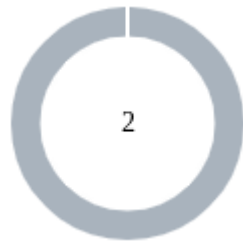
Source Files

Filename	SHA256
MaxYieldUSDT.sol	159f5b17a86c024118dfa20ef22d95f2433 7f9e9afd4acfed834c793e3bf9ba6
lib/DataTypees.sol	6d0d35cfb827b19071232120881ffdddf21 9399511991e303d7a5e19d1d8bfe3
interfaces/IPool.sol	6d93d494ffb95aaf04de797e5c074796f9e8 4eae559e3cc9bc3c1e480265fdc8
interfaces/IERC20Burnable.sol	5ae485cd1b149f2f7300d3fcdcf753d0f0bf c5464fc384ba332fb0cdc624120c

Overview

The `MaxYieldUSDT` contract serves as a unified interface for interacting with the `Hypurrfi` and `Hyperlend` lending protocols, allowing users to efficiently supply and withdraw `USDT`. Leveraging OpenZeppelin's `Ownable` and `Pausable` libraries, it enforces owner control, secure token transfers, and pause functionality. All deposits and withdrawals are handled through a single `execute` function, which routes actions to the appropriate protocol-specific handlers while performing token validation and ownership checks.

Findings Breakdown



● Critical	0
● Medium	0
● Minor / Informative	2

Severity	Unresolved	Acknowledged	Resolved	Other
● Critical	0	0	0	0
● Medium	0	0	0	0
● Minor / Informative	0	2	0	0

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	CCR	Contract Centralization Risk	Acknowledged
●	UTPD	Unverified Third Party Dependencies	Acknowledged

CCR - Contract Centralization Risk

Criticality	Minor / Informative
Location	MaxYieldUSDT.sol#L155,159,163
Status	Acknowledged

Description

The contract's functionality and behavior are heavily dependent on external parameters or configurations. While external configuration can offer flexibility, it also poses several centralization risks that warrant attention. Centralization risks arising from the dependence on external configuration include Single Point of Control, Vulnerability to Attacks, Operational Delays, Trust Dependencies, and Decentralization Erosion.

Shell

```
function pause() public onlyOwner { _pause(); }  
function unpause() public onlyOwner { _unpause(); }  
function withdrawERC20(address token, address to, uint256  
amount) public onlyOwner {  
    IERC20Burnable(token).safeTransfer(to, amount); }
```

Recommendation

To address this finding and mitigate centralization risks, it is recommended to evaluate the feasibility of migrating critical configurations and functionality into the contract's codebase itself. This approach would reduce external dependencies and enhance the contract's self-sufficiency. It is essential to carefully weigh the trade-offs between external configuration flexibility and the risks associated with centralization.

Team Update

The team has acknowledged that this is not a security issue and states:

The contract is meant to be managed by KOPS agent systems to achieve superior yield.

UTPD - Unverified Third Party Dependencies

Criticality	Minor / Informative
Location	MaxYieldUSDT.sol#L56
Status	Acknowledged

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

Shell

```
hypurrfiPool = IPool(initHypurrfiPool);  
hyperlendPool = IPool(initHyperlendPool);  
usdt = IERC20Burnable(initUsdtAddress);  
hypurrfiAToken = IERC20Burnable(initHypurrfiAToken);  
  
hyperlendAToken = IERC20Burnable(initHyperlendAToken);
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

Team Update

The team has acknowledged that this is not a security issue and states:

The external contracts used are from audited protocols of Hypurrfi and Hyperlend.

Functions Analysis


Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
MaxYieldUSDT	Implementation	Pausable, Ownable		
		Public	✓	Ownable
	execute	Public	✓	whenNotPaused
	_hypurrfiSupplyUSDT	Internal	✓	
	_hypurrfiWithdrawUSDT	Internal	✓	
	_hyperlendSupplyUSDT	Internal	✓	
	_hyperlendWithdrawUSDT	Internal	✓	
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	withdrawERC20	Public	✓	onlyOwner

Inheritance Graph



Flow Graph

The flow graph of kops can be found in the following link:

 kops_flow_graph.png

Summary

kops contract implements a yield aggregation mechanism. This audit investigates security issues, business logic concerns and potential improvements. The team has acknowledged all the findings.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does 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. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment 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. Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security. Cyberscope'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. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a TAC blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.



A **TAC Security** Company

The Cyberscope team

cyberscope.io