Security Code Review for YOP Protocol EVM V2

Pluto Digital

April 2022
Version 1.0

Presented by:

BTblock, a FYEO company

Corporate Headquarters

FYEO Inc.

PO Box 147044
Lakewood, CO 80214
United States

Security Level

Strictly Confidential

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Overview

Pluto Digital engaged BTblock LLC to perform a Security Code Review for YOP Protocol EVM V2.

The assessment was conducted remotely by the BTblock Security Team. Testing took place on April 12 - April 20, 2022, and focused on the following objectives:

- Provide the customer with an assessment of their overall security posture and any risks that were discovered within the environment during the engagement.
- To provide a professional opinion on the maturity, adequacy, and efficiency of the security measures that are in place.
- To identify potential issues and include improvement recommendations based on the result of our tests.

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the BTblock Security Teams took to identify and validate each issue, as well as any applicable recommendations for remediation.

Key Findings

The following are the major themes and issues identified during the testing period. These, along with other items, within the findings section, should be prioritized for remediation to reduce to the risk they pose.

{%- for finding in findings %}

- BT-YOP-EVMV2-01 – Missing zero address validation
- BT-YOP-EVMV2-02 – Missing inheritance
- BT-YOP-EVMV2-03 – Unsigned integer is checked to be positive

During the test, the following positive observations were noted regarding the scope of the engagement:

- The team was very supportive and open to discuss the design choices made

Based on formal verification we conclude that the reviewed code implements the documented functionality.

Scope and Rules of Engagement

BTblock performed a Security Code Review for YOP Protocol EVM V2. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

The source code was supplied through a private repository at https://github.com/plutodigital/yop-protocol-evm with the commit hash dd8fc39936b531074749fbc9e59c9ac1547dc7ee. A re-review was performed on April 26, 2022, with the commit hash c94e4442a3a60ac79af33c344e41830d81b1dca5.

F, iles included in the code review
yop-protocol-evm-v2/ ├── changelogs/ │   └── v2.md ├── contracts/ │   ├── access/ │   │   ├── AccessControlManager.sol │   │   ├── AllowAnyAccessControl.sol │   │   ├── AllowListAccessControl.sol │   │   ├── ERC1155AccessControl.sol │   │   ├── PerVaultGatekeeper.sol │   │   └── SanctionsListAccessControl.sol │   ├── fees/ │   │   └── FeeCollection.sol │   ├── interfaces/ │   │   ├── chainalysis/ │   │   │   └── ISanctionsList.sol │   │   ├── convex/ │   │   │   ├── IConvexDeposit.sol │   │   │   └── IConvexRewards.sol │   │   ├── curve/ │   │   │   ├── ICurveAddressProvider.sol │   │   │   ├── ICurveDeposit.sol │   │   │   ├── ICurveGauge.sol │   │   │   ├── ICurveMinter.sol │   │   │   └── ICurveRegistry.sol │   │   ├── roles/ │   │   │   └── IGatekeeperable.sol │   │   ├── sushiswap/ │   │   │   └── IUniswapV2Router.sol │   │   ├── IAccessControlManager.sol │   │   ├── IAccessControlPolicy.sol │   │   ├── IBlockControlPolicy.sol │   │   ├── ICustomHealthCheck.sol │   │   ├── IFeeCollection.sol │   │   ├── IHealthCheck.sol │   │   ├── IStaking.sol │   │   ├── IStrategy.sol │   │   ├── IVault.sol

```
│   │   ├── IVaultStrategyDataStore.sol
│   │   ├── IWeth.sol
│   │   ├── IYOPRegistry.sol
│   │   └── IYOPRewards.sol
│   ├── libraries/
│   │   ├── ConvertUtils.sol
│   │   ├── SwapUtils.sol
│   │   └── VaultUtils.sol
│   ├── mocks/
│   │   ├── strategies/
│   │   │   └── convexv2/
│   │   │       ├── ConvexERC20SinglePoolMock.sol
│   │   │       └── ConvexETHSinglePoolMock.sol
│   │   ├── BasePauseableUpgradeableMock.sol
│   │   ├── BaseStrategyMock.sol
│   │   ├── BaseUpgradeableMock.sol
│   │   ├── BaseVaultMock.sol
│   │   ├── ConvexBtcStrategyMock.sol
│   │   ├── ConvexCurveMetaMock.sol
│   │   ├── ConvexEthStrategyMock.sol
│   │   ├── ConvexStableStrategyMock.sol
│   │   ├── CurveBtcStrategyMock.sol
│   │   ├── CurveERC20SinglePoolMock.sol
│   │   ├── CurveETHSinglePoolMock.sol
│   │   ├── CurveEthStrategyMock.sol
│   │   ├── CurveMetaStrategyMock.sol
│   │   ├── CurveStableStrategyMock.sol
│   │   ├── CustomHealthCheckMock.sol
│   │   ├── HealthCheckMock.sol
│   │   ├── SingleAssetVaultV2BoostedMock.sol
│   │   ├── SingleAssetVaultV2Mock.sol
│   │   ├── StakingMock.sol
│   │   ├── StakingV1Mock.sol
│   │   ├── StakingV1Mock2.sol
│   │   ├── StakingV2Mock.sol
│   │   ├── StrategyMock.sol
│   │   ├── TestnetStrategyMock.sol
│   │   ├── TokenMock.sol
│   │   ├── YOPRewardsMock.sol
│   │   ├── YOPRewardsV2Mock.sol
│   │   ├── YOPTokenMock.sol
│   │   └── YopERC1155Mock.sol
│   ├── registry/
│   │   └── YOPRegistry.sol
│   ├── rewards/
│   │   ├── YOPRewards.sol
│   │   └── YOPRewardsV2.sol
│   ├── router/
│   │   └── YOPRouter.sol
│   ├── security/
│   │   ├── BasePauseableUpgradeable.sol
│   │   └── BaseUpgradeable.sol
│   ├── staking/
│   │   ├── Staking.sol
│   │   └── StakingV2.sol
│   ├── strategies/
│   │   ├── convexv2/
│   │   │   ├── ConvexCurveMeta.sol
│   │   │   ├── ConvexERC20SinglePool.sol
│   │   │   └── ConvexETHSinglePool.sol
│   │   ├── curvev2/
│   │   │   ├── CurveBaseV2.sol
│   │   │   ├── CurveERC20SinglePool.sol
│   │   │   ├── CurveETHSinglePool.sol
│   │   │   └── CurveMeta.sol
│   │   ├── BaseStrategy.sol
│   │   ├── ConvexBase.sol
│   │   ├── ConvexBtc.sol
│   │   ├── ConvexEth.sol
│   │   ├── ConvexStable.sol
│   │   ├── CurveBase.sol
│   │   ├── CurveBtc.sol
│   │   ├── CurveEth.sol
│   │   └── CurveStable.sol
│   └── vaults/
│       ├── roles/
│       │   ├── Gatekeeperable.sol
│       │   ├── Governable.sol
│       │   └── Manageable.sol
│       ├── BaseVault.sol
│       ├── CommonHealthCheck.sol
│       ├── SingleAssetVault.sol
│       ├── SingleAssetVaultBase.sol
│       ├── SingleAssetVaultV2.sol
│       ├── VaultDataStorage.sol
│       ├── VaultMetaDataStore.sol
│       └── VaultStrategyDataStore.sol
├── deployment-config/
│   ├── config-example.yaml
│   ├── config-rinkeby-staging.yaml
│   ├── config-test-rinkeby.yaml
│   ├── mainnet-production.yaml
│   └── vaults.ts
├── deployments/
│   ├── mainnet-production.json
│   ├── rinkeby-staging.json
│   ├── rinkeby.json
│   └── test-rinkeby.json
├── flat/
│   ├── AccessControlManager_flat.sol
│   ├── Inbox_flat.sol
│   └── SingleAssetVault_flat.sol
├── scripts/
│   ├── gnosis/
│   │   ├── propose-txn.ts
│   │   ├── safe-create.ts
│   │   └── safe-delegate.ts
│   ├── lib/
│   │   ├── AccessControlManagerDeployment.ts
│   │   ├── AllowAnyAccessControlDeployment.ts
│   │   ├── AllowlistAccessControlDeployment.ts
│   │   ├── ContractDeployment.ts
│   │   ├── ConvexStrategyDeployment.ts
│   │   ├── CurveStrategyDeployment.ts
│   │   ├── ERC1155AccessControlDeployment.ts
│   │   ├── Executor.ts
│   │   ├── FeeCollectionDeployment.ts
│   │   ├── MockStrategyDeployment.ts
│   │   ├── StakingDeployment.ts
│   │   ├── VaultDeployment.ts
│   │   ├── VaultStrategyDataStoreDeployment.ts
│   │   └── YopRewardDeployment.ts
│   ├── README.md
│   ├── deploy-all.ts
│   ├── deploy-by-config.ts
│   ├── deploy-contract.ts
│   ├── deploy-mock.ts
│   ├── propose-upgrade.ts
│   ├── util.ts
│   └── verify.ts
├── tasks/
│   ├── fork/
│   │   ├── fundAccounts.ts
│   │   ├── impersonateAccounts.ts
│   │   └── reset.ts
│   ├── gnosis/
│   │   ├── propose-txn.ts
│   │   └── safe-create.ts
│   ├── kms/
│   │   └── kms-eth-address.ts
│   ├── rewards/
│   │   └── approveRewardsContract.ts
│   └── index.ts
├── README.md
├── commitlint.config.js
├── constants.ts
├── deploy-to-mainnet.md
├── hardhat.config.ts
├── package-lock.json
├── package.json
├── tenderly.yaml
└── tsconfig.json
```

Table 1: Scope

Technical Analyses and Findings

During the Security Code Review for YOP Protocol EVM V2, we discovered:

- 1 finding with LOW severity rating.
- 2 findings with INFORMATIONAL severity rating.

The following chart displays the findings by severity.

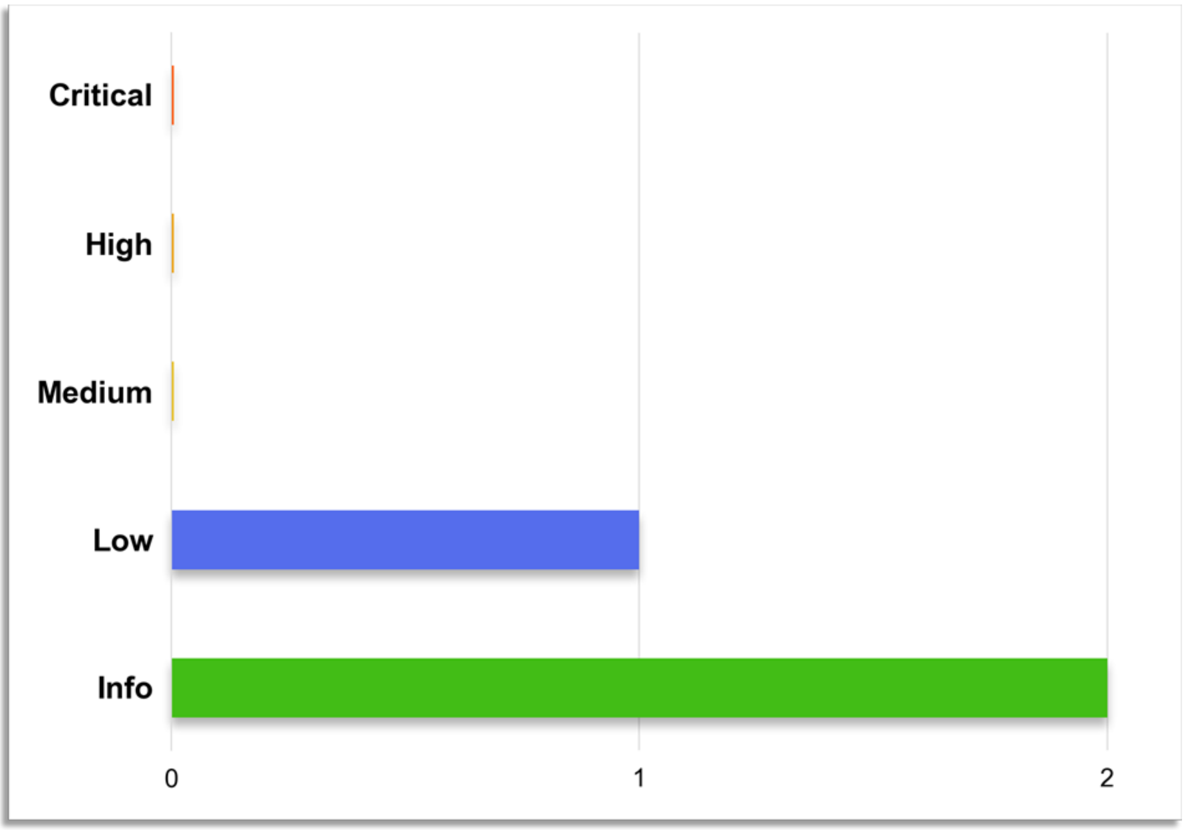

Figure 1: Findings by Severity

Findings

The Findings section provides detailed information on each of the findings, including methods of discovery, explanation of severity determination, recommendations, and applicable references.

The following table provides an overview of the findings.

Finding #	Severity	Description
KS-YOP-EVMV2-01	Low	Missing zero address validation
KS-YOP-EVMV2-02	Informational	Missing inheritance
KS-YOP-EVMV2-03	Informational	Unsigned integer is checked to be positive

Table 2: Findings Overview

Technical Analyses

Based on the source code, the validity of the code was verified and confirmed that the intended functionality was implemented correctly and to the extent that the state of the repository allowed, unless otherwise stated.

Based on formal verification we conclude that the code implements the documented functionality to the extent of the reviewed code.

Technical Findings

General Observations

The Yield Optimization Platform (YOP) V2 introduces upgraded contracts that adds support for boosted APY. This update will result in an increased reward for users who have staked YOP tokens. The following contracts were added for V2:

- vaults/SingleAssetVaultV2.sol
- rewards/YOPRewardsV2.sol
- staking/StakingV2.sol
- router/YOPRouter.sol
- registry/YOPRegistry.sol
- access/SanctionsListAccessControl.sol
- strategies/curvev2/
- strategies/convexv2/

Missing zero address validation

Finding ID: KS-YOP-EVMV2-01
Severity: Low
Status: Remediated

Description

Some setter functions do not check for zero addresses.

Proof of Issue

File name: FeeCollection.sol
Line number: 181

```
    function setProtocolWallet(address _protocolWallet) external onlyGovernance {
       protocolWallet = _protocolWallet;
    }
```

File name: VaultMetaDataStore.sol
Line number: 114

```
    function setVaultCreator(address _creator) external {
      _onlyGovernanceOrGatekeeper(governance);
      creator = _creator;
    }
```

File name: YOPRewards.sol
Line number: 363

```
    function setRewardWallet(address _wallet) external onlyGovernance {
      rewardsWallet = _wallet;
    }
```

Severity and Impact Summary

Zero address may lead to financial losses in the case if assets are transferred to it.

Recommendation

It is recommended to check that the address is not zero.

Missing inheritance

Finding ID: KS-YOP-EVMV2-02
Severity: Informational
Status: Remediated

Description

Some classes implement interface functionality without actual inheritance.

Proof of Issue

`FeeCollection` does not inherit `IFeeCollection`.

File name: FeeCollection.sol
Line number: 13

```
    contract FeeCollection is BasePauseableUpgradeable {
```

`CommonHealthCheck` does not inherit `IHealthCheck`.

File name: CommonHealthCheck.sol
Line number: 14

```
    contract CommonHealthCheck {
```

Severity and Impact Summary

Inheritance may help the compilator to find inconsistencies between interface and implementation.

Recommendation

It is recommended to add inheritance from interfaces.

Unsigned integer is checked to be positive

Finding ID: KS-YOP-EVMV2-03
Severity: Informational
Status: Remediated

Description

An unsigned integer could not be less than zero, so the expression >= 0 is meaningless.

Proof of Issue

File name: FeeCollection.sol
Line number: 158

```
    require(_ratio >= 0 && _ratio <= MAX_BPS, "!ratio");
```

File name: FeeCollection.sol
Line number: 342

```
    require(_proposerRatio >= 0 && _proposerRatio <= MAX_BPS, "!ratio");
```

File name: FeeCollection.sol
Line number: 343

```
    require(_developerRatio >= 0 && _developerRatio <= MAX_BPS, "!ratio");
```

File name: FeeCollection.sol
Line number: 351

```
    require(_ratio >= 0 && _ratio <= MAX_BPS, "!ratio");
```

Severity and Impact Summary

The code that does not impact the logic just increases the cost of a transaction in GAS.

Recommendation

It is recommended to remove excessive comparison or to change the value type.