(https://blog.coinfabrik.com/)

SMART CONTRACT AUDIT (HTTPS://BLOG.COINFABRIK.COM/CATEGORY/SMART-CONTRACTS/SMART-CONTRACT-AUDIT-SMART-CONTRACTS/)

YFFII Smart Contract Audit

Mariana Soffer (https://blog.coinfabrik.com/author/mariana-soffel/)

October 21, 2020 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/yffii-smart-contract-audit/)

Contents

Introduction

CoinFabrik was asked to audit the contracts for the YFFII project. First we will provide a summary of our discoveries and then we will show the details of our findings.

Summary

The contracts audited are the contracts deployed at 0x91a06884F6db45FF499cC2f7C6f3eb60a617D5AE (https://etherscan.io/address/0x91a06884f6db45ff499cc2f7c6f3eb60a617d5ae) and 0x900E9bAEc63BcdB9FBb0D9743326360e6AE4B2dB (https://etherscan.io/address/0x900E9bAEc63BcdB9FBb0D9743326360e6AE4B2dB).

Contracts

The audited contracts are:

- *Controller.sol (md5: 304e66a01ff2b10b81763a06f09bc6e2)*
- *YearnRewards.sol (md5: a6937e2f61dcabc8bca2825bd45e485e)*

Analyses

The following analyses were performed:

- Misuse of the different call methods
- Integer overflow errors
- Division by zero errors
- Outdated version of Solidity compiler
- Front running attacks
- Reentrancy attacks
- Misuse of block timestamps
- Softlock denial of service attacks
- Functions with excessive gas cost
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Failure to use a withdrawal pattern
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

Detailed findings

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. They must be fixed **immediately**.

- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.
- **Enhancement:** These kinds of findings do not represent a security risk. They are best practices that we suggest to implement.

This classification is summarized in the following table:

SEVERITY	EXPLOITABLE	ROADBLOCK	TO BE FIXED
Critical	Yes	Yes	Immediately
Medium	In the near future	Yes	As soon as possible
Minor	Unlikely	No	Eventually
Enhancement	No	No	Eventually

Issues Found by Severity

Critical severity

No issues of critical severity found.

Medium severity

No issues of medium severity found

Minor severity

No issues of minor severity found

Enhancements

Outdated Solidity compiler version

The latest Solidity compiler version currently available is 0.7.4, switching from 0.5.x to 0.7.x involves modifying the code however given that the project is still small, we recommend considering to do so as it would allow you to enjoy the latest bug fixes and gas optimization

changes to the date.

If you wish to stay with the 0.5.16 version we recommend not using a floating pragma as it allows older versions of 0.5.x to be used, the following pragma is more appropriate: pragma solidity 0.5.16;

Commonly used require statements can be changed to modifiers

Currently the following require statement can be seen in many functions through the code:

```
require(msg.sender == governance, "!governance");
```

This can be refactored into using a modifier, so that only an extra word needs to be used in each function header when needed:

```
modifier onlyGovernance {
 require(msg.sender == governance, "!governance");
 _;
}

[···]

function setRewards(address _rewards) public onlyGovernance {
        rewards = _rewards;
}
```

The same can be done with the following require statement:

```
require(msg.sender == strategist || msg.sender == governance, "!governance");
```

Like this:

```
modifier onlyGovernanceOrStrategist {
 require(msg.sender == strategist || msg.sender == governance, "!governance");
 _;
}

[···]

function inCaseTokensGetStuck(address _token, uint _amount) public onlyGovernanceOrStrategis
t {
    IERC20(_token).safeTransfer(msg.sender, _amount);
}
```

This change is merely for reducing code complexity and does not change the logic in any way.

Conclusion

We found the contracts to be simple and straightforward and to have an overall good quality code.

A large part of the logic is done in external contracts, it is very important that you make sure to always use the correct addresses referring to trusted and verified contracts, and that these can not get changed by external parties.

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the YFFII project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee. External contracts have not been audited by CoinFabrik and were assumed to work properly during this audit.

Related Posts

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/hex-smart-contract-audit/)

HEX Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/hex-smart-contract-audit/)

CoinFabrik was asked to audit the contracts for the HEX project. Firstly, we will provide…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/icherry-smart-contract-audit/)

ICherry Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/icherry-smart-contract-audit/)

CoinFabrik was asked to audit the contracts for the ICherry Finance project. First we will…

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/katana-smart-contract-audit/)

Katana Smart Contract Audit

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/katana-smart-contract-audit/)

CoinFabrik was asked to audit the contracts for the Katana project. First we will provide...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/securitize-smart-contract-audit/)

Securitize Smart Contract Audit

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/securitize-smart-contract-audit/)

CoinFabrik was asked to audit the Securitize contracts for the DSToken project. First we will...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/timvi-smart-contract-audit/)

Timvi Smart Contract Audit

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/timvi-smart-contract-audit/)

CoinFabrik was asked to audit the contracts for the Timvi project. Firstly, we will provide...

contract-
audit/)

(https://blog.coinfabrik.com/smart-
contracts/smart-
contract-
audit-
smart-
contracts/katana-
smart-
contract-
audit/)

Katana Smart Contract Audit
(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/katana-smart-contract-audit/)
CoinFabrik was asked to audit the contracts for the Katana project. First we will provide…

(https://blog.coinfabrik.com/smart-
contracts/etherparty-
token-
smart-
contract-
security-
audit-
coinfabrik/)

EtherParty Smart Contract Security Audit
(https://blog.coinfabrik.com/smart-contracts/etherparty-token-smart-contract-security-audit-coinfabrik/)
Coinfabrik team has been hired to audit Etherparty smart contracts. Firstly, we will provide a…

(https://blog.coinfabrik.com/smart-
contracts/smart-
contract-
audit-

Stasis Token Smart Contract Audit
(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/stasis-token-smart-contract-audit/)

smart-contracts/stasis-token-smart-contract-audit/)

Coinfabrik has been hired to audit the smart contracts for Stasis sale, the Stable Euro...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-basiccosigner-smart-contract-security-audit/)

RCN BasicCosigner Smart Contract Security Audit

Coinfabrik security audit's team was asked to audit the BasicCosigner contract. This contract is part...

Tags:

security audit (https://blog.coinfabrik.com/tag/security-audit/)

Smart Contract (https://blog.coinfabrik.com/tag/smart-contract/)

smart contract audit (https://blog.coinfabrik.com/tag/smart-contract-audit/)

yield farm (https://blog.coinfabrik.com/tag/yield-farm/)

yield farming (https://blog.coinfabrik.com/tag/yield-farming/)

SHARE ON

(https://www.facebook.com/sharer/sharer.php?u=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/yffii-smart-contract-audit/)
(https://twitter.com/intent/tweet?text=YFFII%20Smart%20Contract%20Audit&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/yffii-smart-contract-audit/)
(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-content/uploads/2020/10/images.png&description=YFFII+Smart+Contract+Audit)
(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/yffii-smart-contract-audit/&title=YFFII%20Smart%20Contract%20Audit&source=CoinFabrik%20Blog)

← PREVIOUS ARTICLE

Wise Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/wise-smart-contract-audit/)

NEXT ARTICLE →

1Inch Exchange Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/1inch-exchange-smart-contract-audit/)

You may also like

(https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

Magic Bridge Audit (https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

2 months ago

Smart Contract Audit (https://blog.coinfabrik.com/category/smart-contracts/smart-contract-audit-smart-contracts/)

MintingFactoryV2, BaseUpgradableMarketplace & KODAV3UpgradableGatedMarketplace (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

(https://blog.coinfabrik.com/feed/)

(https://ar.linkedin.com/company/coinfabrik)

(https://twitter.com/coinfabrik)

(https://www.youtube.com/channel/UC2GmjCr7aEz-il31kqOy9aw)

(https://www.facebook.com/CoinFabrik/)

(https://www.reddit.com/r/CoinFabrik/)

(https://github.com/coinfabrik)

Made with ♥ in Buenos Aires, Argentina.