HackSafe

Smart Contract Security Audit Report

PolyYeld Token

November 2022

Security Status

★★★★⯪

www.hacksafe.io

Audit Details

Audited project
PolyYeld Token

Deployer address
0x49854708a8c42eeb837a97dd97d597890ceb1334

Client contacts
PolyYeld Token Team

Blockchain
Polygon

Website
https://polyyeld.finance/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: 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 any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and HackSafe 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 have effect in relation to the report. Except and only to the extent that it is prohibited by law, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by PolyYeld Token to perform an audit of smart contracts:

- https://polygonscan.com/address/0xd0f3121A190d85dE0AB6131f2bCEcdbfcfB38891#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contract Details

Token contract details for 23.11.2022

Token Type	: DEFI
Contract name	: POLYYELD
Contract address	: 0xd0f3121A190d85dE0AB6131f2bCEcdbfcfB38891
Total supply	: 4,995,853,271,702.385986
Token ticker	: YELD
Decimals	: 18
Token Holders	: 4,792
Transactions count	: 5,381,774
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0x49854708a8c42eeb837a97dd97d597890ceb1334
Owner address	: 0x2dc11b394bd0f1cc6ac0a269cfe3cc0b333601b4

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secure".** This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here

We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 1 low.

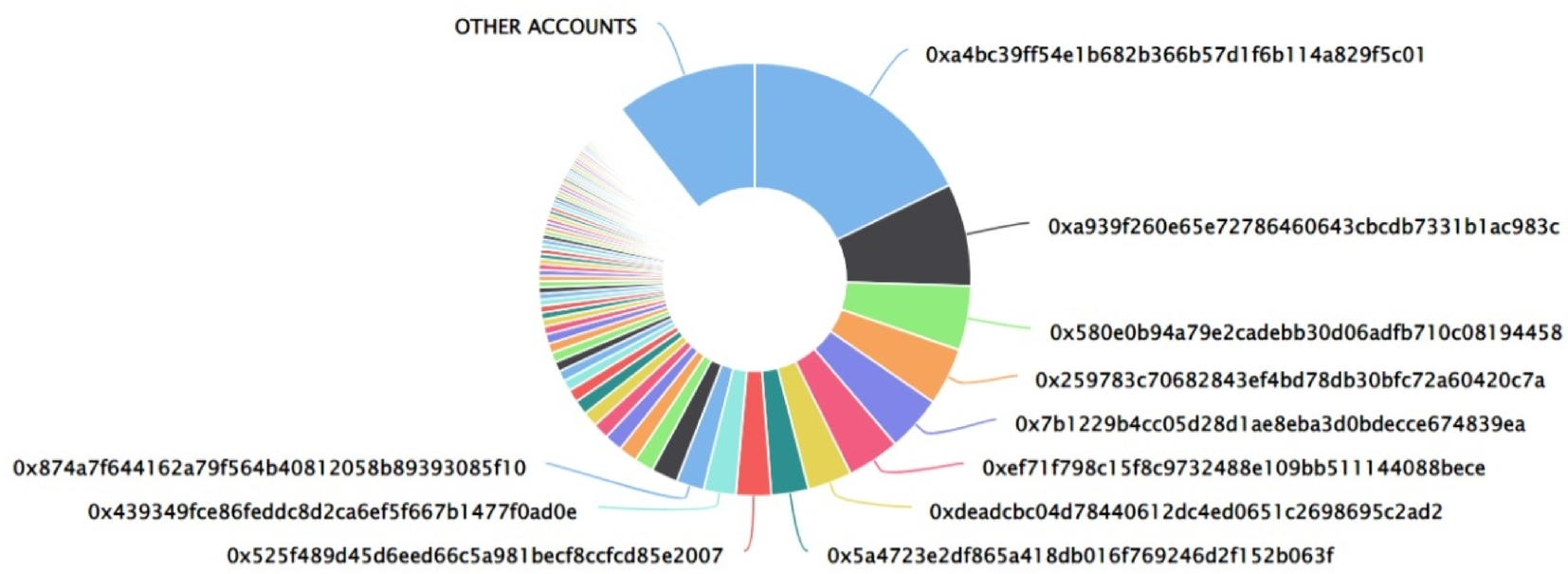
PolyYeld Token Distribution

The top 100 holders collectively own 89.41% (4,466,718,249,900.71 Tokens) of PolyYeld Token

Token Total Supply: 4,995,853,271,702.39 Token | Total Token Holders: 4,792

PolyYeld Token Top 100 Token Holders

Source: polygonscan.com



PolyYeld Token Top 20 Token Holders

(A total of 4,466,718,249,900.71 tokens held by the top 100 accounts from the total supply of 4,995,853,271,702.39 token)

Rank	Address	Quantity (Token)	Percentage
1	0xa4bc39ff54e1b682b366b57d1f6b114a829f5c01	890,000,000,000	17.8148%
2	0xa939f260e65e72786460643cbcdb7331b1ac983c	382,647,970,252.791989708173592767	7.6593%
3	0x580e0b94a79e2cadebb30d06adfb710c08194458	239,973,833,050.304767008800415358	4.8035%
4	0x259783c70682843ef4bd78db30bfc72a60420c7a	214,634,996,439.605913527621304349	4.2963%
5	0x7b1229b4cc05d28d1ae8eba3d0bdecce674839ea	208,469,074,061.874461925181149176	4.1728%
6	0xef71f798c15f8c9732488e109bb511144088bece	197,127,611,666.520181205549715471	3.9458%
7	0xdeadcbc04d78440612dc4ed0651c2698695c2ad2	165,813,983,843.765512099359615003	3.3190%
8	0x5a4723e2df865a418db016f769246d2f152b063f	139,285,182,404.554591478767095825	2.7880%
9	0x525f489d45d6eed66c5a981becf8ccfcd85e2007	130,589,965,238.924673072206701624	2.6140%
10	0x439349fce86feddc8d2ca6ef5f667b1477f0ad0e	121,327,082,210.01983514048056463	2.4286%
11	0x874a7f644162a79f564b40812058b89393085f10	103,408,581,351.046231674928622692	2.0699%
12	0x8dc7bf3baf5c8ae60b14daeb2bcde95b32d94a84	98,010,343,814.721873251589159076	1.9618%
13	0xabe27983bcce8bc9db2089ce104a1876bc701700	72,901,150,403.625473524091816977	1.4592%
14	0x7a0876100fafa869ce35b30e4d99727bf4735645	69,574,799,692.726041096410806229	1.3927%
15	0xa679380e7c781907f94ad06c0b1595bf83820666	66,631,794,974.879997091820970043	1.3337%
16	0xcf0a6afe5320654037fd5877f9cec0656d594e70	60,213,374,471.763871879396147584	1.2053%
17	0xc6d8505a6ed05c0ff895127e49089d911ee03882	57,265,205,915.086766136397859811	1.1463%
18	0xd786f7f6a3a37c42fcb90529f85856b0b5a41872	53,886,762,769.786090102973235913	1.0786%
19	0xb613d0a877a2d7300eb74e517a4e6d0802f97537	47,060,376,143.224414422952054964	0.9420%
20	0x70d84adf0fc5a6764d495d95d57714ca26c095bb	40,000,000,000	0.8007%

PolyYeld Token Distribution

PolyYeld Token Contract Overview

Contract functions details

```
+[Lib] SafeMath
    -[Int] add
    -[Int] sub
    -[Int] sub
    -[Int] mul
    -[Int] div
    -[Int] div
    -[Int] mod
    -[Int] mod

+[Int] IBEP20
    -[Ext] totalSupply
    -[Ext] decimals
    -[Ext] symbol
    -[Ext] name
    -[Ext] getOwner
    -[Ext] balanceOf
    -[Ext] transfer
    -[Ext] allowance
    -[Ext] approve
    -[Ext] transferFrom

+Context
    -[Int] _msgSender
    -[Int] _msgData

+Ownable (Context)
    -[Pub] <constructor>
    -[Pub] owner
    -[Pub] renounceOwnership #
      -modifiers: onlyOwner
    -[Pub] transferOwnership #
      -modifiers: onlyOwner

+ BEP20 (Context, IBEP20, Ownable)
    -[Pub] <constructor>
    -[Ext] getOwner
    -[Pub] name
    -[Pub] name
```

Contract functions details

-[Pub] decimals
-[Pub] totalSupply
-[Pub] maxSupply
-[Pub] balanceOf
-[Pub] transfer #
-[Pub] allowance
-[Pub] approve #
-[Pub] transferFrom #
-[Pub] increaseAllowance #
-[Pub] decreaseAllowance #
-[Pub] mint #
  - modifiers: onlyOwner
-[Int] _transfer #
-[Int] _mint #
-[Int] _burn #
-[Int] _approve #
-[Int] _burnFrom #

+ POLYYELD (BEP20)
  -[Pub] mint #
    - modifiers: onlyOwner
  -[Ext] delegates
  -[Ext] delegate #
  -[Ext] delegateBySig #
  -[Ext] getCurrentVotes
  -[Ext] getPriorVotes
  -[Int] _delegate #
  -[Int] _moveDelegates #
  -[Int] _writeCheckpoint #
  -[Int] safe32
  -[Int] getChainId

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Low issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

- **Critical Severity Issues**
  No critical severity issue found.
- **High Severity Issues**
  No high severity issue found.
- **Medium Severity Issues**
  No medium severity issue found.
- **Low Severity Issues**
  One low severity issue found.

1. Old compiler version

- **Description**
  Contract has been deployed using too old solidity version.
- **Recommendation**
  It is advisable to deploy contract using any of the latest version of solidity

Centralization

Owner Privileges :

- PolyYeld Token Contract:
    - Owner can renounce and transfer ownership.
    - Owner can mint tokens.

This smart contract has some functions which can be executed by the admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble as smart contract ownership has not been renounced. Following are Admin functions:

- renounceOwnership
- transferOwnership
- mint

Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.