SECURITY AUDIT OF

PENTAGON ERC20 TOKEN

Public Report

Nov 14, 2024

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Nov 14, 2024. We would like to thank the Pentagon for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Pentagon ERC20 Token. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified some vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY 5

1.1. About Pentagon ERC20 Token 5

1.2. Audit Scope 5

1.3. Audit Methodology 5

1.4. Disclaimer 6

1.5. Acceptance Minute 7

2. AUDIT RESULT 8

2.1. Overview 8

2.2. Findings 8

2.3. Draft Findings 8

2.3.1. Wrong logic in `transferAndCall()` function. HIGH 9

2.3.2. Not using the some logic in the contracts. INFORMATIVE 9

3. VERSION HISTORY 11

1. MANAGEMENT SUMMARY

1.1. About Pentagon ERC20 Token

Pentagon is a multi-platform blockchain project that creates a decentralized ecosystem for gaming and entertainment applications. This project aims to combine blockchain technology with the gaming industry to provide a transparent and secure experience for players, while offering developers a flexible platform for deploying interoperable games. Some of these parts of Pentagon are Pentagon ZkEVM, Pentagon Chain Apps, Hexagon city and Pentagon ecosytem games and IPS.

1.2. Audit Scope

This audit focused on identifying security flaws in code and the design of the Pentagon ERC20 Token.

SHA256 Sum	File
8a3791cada14c6bb48cd228cd07fd6760d50bf763db529b11272184d4016aac0	contracts/PEN.sol
7ec3680373c0cde8e286aef284b820308f1ae4ffde81077958fd883c92b1791f	contracts/Freezable.sol
a3b7ca552ad1f7ab2d3ba0fc4f9b42b8d234a08a3e49aa18976258c87d66caf9	contracts/shared/BasicAccessControl.sol
acdea85c9fbf8263e9b0decb9f15678676fafcc5d9425a6e9b1d964d6f1d5af6	contract/interface/IERC677Receiver.sol

1.3. Audit Methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence

- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

Pentagon acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. Pentagon understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, Pentagon agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the Pentagon will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the Pentagon, the final report will be considered fully accepted by the Pentagon without the signature.

2. AUDIT RESULT

2.1. Overview

The Pentagon ERC20 Token was written in `Solidity` language.

The contracts extend `ERC20`, `ERC20Permit`, and `ERC20Burnable` from the `OpenZeppelin` library. The contracts also include the `IERC677Receiver` interface. The `PEN` contract is a custom ERC20 token with `toggleIsTransferable`, `_beforeTokenTransfer`, freezing mechanism, and access control mechanism.

Moderators can freeze any account with `freeze` then that user can not transfer or receive tokens. Besides, the tokens can only be transfered when `isTransferable` is True. Only owner can set the `isTransferable`. Owner can also add or remove moderator by `addModerator`.

Below table describes some properties of the audited Pentagon ERC20 Token (as of the report writing time):

PROPERTY	VALUE
Name	Pentagon Token
Symbol	PEN
Decimals	18
Total Supply	1,000,000,000 (x10^{18}) Note: the number of decimals is 18, so the total representation token will be 1,000,000,000 or 1 billion.

Table 2. Pentagon Token's properties

2.2. Findings

2.3. Draft Findings

#	Title	Severity	Status
1	Wrong logic in `transferAndCall()` function.	HIGH	Fixed
2	Not using the some logic in the contracts.	INFORMATIVE	Fixed

2.3.1. Wrong logic in `transferAndCall()` function. HIGH

Position

- `PEN.sol`#transferAndCall()

Description

This function uses `IERC677Receiver`. The `transferAndCall` function can be called to transfer tokens to a contract and then call the contract with the additional data provided. Once the token is transferred, the token contract calls the receiving contract's function `onTokenTransfer(address,uint256,bytes)` and triggers an event. But the `transferAndCall()` function in this contract calls `onTokenTransfer` before transferring tokens to the contract. In additional, `ERC677` is an unapproved `EIP`.

```
function transferAndCall(
        address _to,
        uint256 _value,
        bytes memory _data
    ) public returns (bool success) {
        if (isContract(_to)) {
            IERC677Receiver receiver = IERC677Receiver(_to);
            receiver.onTokenTransfer(msg.sender, _value, _data);
        }

        transfer(_to, _value);
        emit Transfer(msg.sender, _to, _value, _data);
        return true;
    }
```

RECOMMENDATION

Checking `ERC1363` which has been inspired by `ERC677` and went through the EIP process. Reference links: EIP-ERC1363, Openzeppelin-ERC1363-contract.

UPDATES

The Pentagon team has been acknowledged the issue. The `transferAndCall` mechanism has been removed by the team.

2.3.2. Not using the some logic in the contracts. INFORMATIVE

Position

- `shared/BasicAccessControl.sol`#isActive()
- `interface/IERC677Receiver.sol`

Description

The `BasicAccessControl.sol` contract defines an `isActive` modifier, but it is currently unused. Additionally, with the removal of the `transferAndCall` mechanism, `IERC677Receiver` is no longer utilized. It would be advisable to review these mechanisms and consider removing them if they are no longer needed.

UPDATES

The Pentagon team has been acknowledged this. The `IERC677Receiver` and `isActive` have been removed by the team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Nov 12, 2024*	Public Report	Verichains Lab
1.1	*Nov 13, 2024*	Public Report	Verichains Lab
1.2	*Nov 14, 2024*	Public Report	Verichains Lab

Table 3. Report versions history