Tetragaze

Smart contract audit

POLKA STATION

Tetragaze

September 15, 2021 | v 2.0

Pass

Tetragaze Security Team has concluded that there are no issues that can have an impact on contract security. The contract is well written and is production-ready.

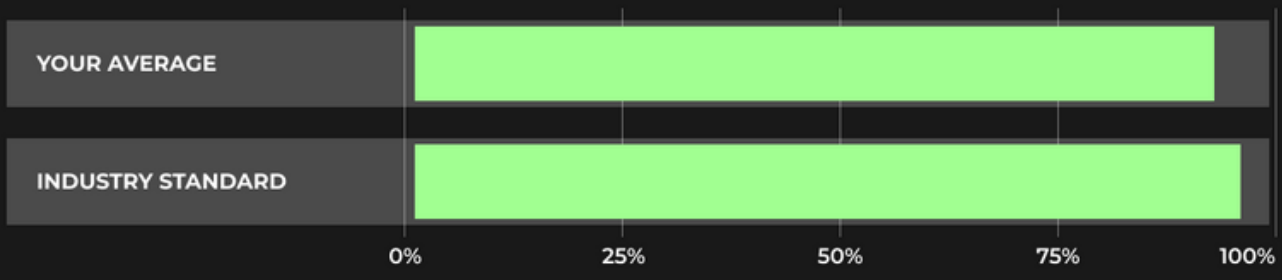
Technical summary

This document outlines the overall security of the PolkaStation smart contracts, evaluated by Tetragaze Blockchain Security team.
The scope of this audit was to analyze and document the PolkaStation smart contract codebase for quality, security, and correctness

Contract Status

There were no critical issues found during the audit.

Testable Code



Testable code is 91%, which is close to the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the BNB Chain network's fast-paced and rapidly changing environment, we at Tetragaze recommend that the PolkaStation team put in place a bug bounty program to encourage further and active analysis of the smart contract.

Table of content

Auditing Strategy and Techniques Applied

Requirements: FIP8 spec

During the review process, we made sure to carefully check that the token contract:

- Adheres to established Token standards and ensures effective implementation;
- Ensures that documentation and code comments accurately reflect the logic and behavior of the code;
- Ensures that token distribution is in line with calculations;
- Utilizes best practices to efficiently use gas, avoiding unnecessary waste;
- Employs methods that are safe from reentrance attacks and immune to the latest vulnerabilities;
- Follows best practices for code readability

Tetragaze team of expert pentesters and smart contract developers carefully evaluated the repository for security issues, code quality, and compliance with specifications and best practices. They conducted a line-by-line review, documenting any issues that were identified during the process. Special care was taken to ensure that the review was thorough and comprehensive.

1	Due diligence in assessing the overall code quality of the codebase.	2	Cross-comparison with other, similar smart contracts by industry leaders.
3	Testing contract logic against common and uncommon attack vectors.	4	Thorough, manual review of the codebase, line-by-line.

Summary

PolkaStation is an intriguing project that boasts a welcoming and expanding community. Its smart contract has been thoroughly analyzed and no critical errors or issues were identified. The contract owner has access to certain administrative functions, but these cannot be misused to disrupt the transactions of users.

Structure and Organization of Document

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the ability of the contract to compile or operate in a significant way.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

Complete Analysis

Public Function could be Declared External

Low

It is recommended to declare public functions that are never called by the contract as external to conserve gas.

```
name
symbol
decimals
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
...
```

Recomendation:

"Functions marked with the external attribute should only be used for functions that are not called within the contract.

Complete Analysis

Redundant Statements

Low

The contract includes unnecessary statements that do not serve any purpose and simply add to the code size. These segments have no impact on the contract and can be removed to streamline the code.

```
Context
```

Recomendation:

Eliminate unnecessary statements to decrease the code size.

Complete Analysis

Dead Code Elimination

Low

Unused functions that increase the size of the contract's code.

```
_burn
```

Recomendation:

Remove unused functions.

Complete Analysis

Variable shadowing in a local scope

Low

There are variables that are defined in the local scope that have the same name as those in an upper scope.

```
_owner
totalSupply
```

Recomendation:

Upper scoped variables should not share the same names as local variables.

We are thankful for the opportunity to collaborate with the PolkaStation team.

This document's statements should not be taken as investment or legal advice, and the authors cannot be held responsible for any decisions made based on them.

It is suggested by the Tetragaze Security Team that the PolkaStation team implement a bug bounty program in order to encourage external parties to scrutinize the smart contract further.