SIZ Technologies Pvt. Ltd.

# GABUR SMART CONTRACT AUDIT

**INTRODUCTION:**

The audit document highlights the syntax, standards, semantics, and security of smart contracts. The report ensures manual and tool-based proper assessment of smart contract code. Osiz team started with analyzing and understanding contract architecture and code design patterns. The following focus is on security flaws, quality, and correctness. Also, performed line by line manual analyses for the potential issue.

**AUDITING METHODOLOGY:**

we perform the audit according to the following procedure:

- **Basic Coding Bugs:** We statically analyze given smart contracts for known coding bugs, and then manually verify (reject or confirm) all the issues.
- **Semantic Consistency Checks:** We then manually check the logic of implemented smart contracts.
- **Advanced Security Checks:** We further review business logic, examine system operations to uncover possible pitfalls and/or bugs.
- **Additional Recommendations:** We also provide additional suggestions regarding the coding and development of smart contracts from the perspective of proven programming practices.

**CRITICAL ISSUES (critical, high severity):**

Bugs and vulnerabilities that enable theft of funds, lock access to funds without possibility to restore it, or lead to any other loss of funds to be transferred to any party; high priority unacceptable bugs for deployment at main network;

**ERRORS, BUGS AND WARNINGS (medium, low severity):**

Bugs that can trigger a contract failure, with further recovery only possible through manual modification of the contract state or contract replacement altogether; Lack of necessary security precautions;

**OPTIMIZATION POSSIBILITIES (very low severity):**

Possibilities to decrease the cost of transactions and data storage of Smart-Contracts.

**NOTES AND RECOMMENDATIONS (very low severity):**

Tips and tricks, all other issues and recommendations, as well as errors that do not affect the functionality of the Smart-Contract.

**CONCLUSION:**

In the Smart-Contracts were found no vulnerabilities and no backdoors. The code was manually reviewed for all commonly known and more specific vulnerabilities. So Smart-Contract is safe for use in the main network.

**SOURCE:**

**Gabur :** https://bscscan.com/address/0x0Bbe4eBC8A16C759eA82D0950f29Aad3E3F10070#code

**AUDIT SUMMARY:**

The code quality is well maintained and is well written following to the solidity coding standards.

**LOW SEVERITY ISSUE:**

**Lock pragmas to specific compiler version :**

Locking the pragma helps ensure that contracts do not accidentally get deployed using latest compiler version as it may have higher risks of undiscovered bugs. Nightly builds should never be used to compile code for production.

```
7  pragma solidity ^0.4.24;
8
```

**Usage:**

pragma solidity ^0.4.24;

**Recommended:**

pragma solidity =0.8.13;

**Visibility modifier :**

Visibility modifier must be first in list of modifiers

```
/**
 * @dev called by the owner to pause, triggers stopped state
 */
function pause() onlyOwner whenNotPaused public {
  paused = true;
  emit Pause();
}
```

**Usage:**

```
function pause() onlyOwner whenNotPaused public {
   paused = true;
   emit Pause();
 }
```

**Recommended:**

```
function pause() public  onlyOwner whenNotPaused {
   paused = true;
   emit Pause();
 }
```

**MEDIUM SEVERITY ISSUE:**

The audit did not find any medium severity issues.

**HIGH SEVERITY ISSUE:**

The audit did not find any high severity issues.