AssetfinX

SMART CONTRACT SECURITY AUDIT REPORT

INTRODUCTION

The audit document highlights the standards, semantics, and security of smart contracts. The report ensures manual and tool-based proper assessment of smart contract code. Assetfinx 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 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 mainnet;

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 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

AFRICUNIA BANK

https://exlscan.com/address/0x8ba1940D299d3fd2d64DEB9BA8c552940A8C5d3b/contracts

AUDIT SUMMARY

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

VULNERABILITY SUMMARY

Total Issues	**1**
High	**0**
Medium	**0**
Low	**1**

AFRICUNIA BANK

HIGH SEVERITY ISSUE

The audit did not find any high severity issues.

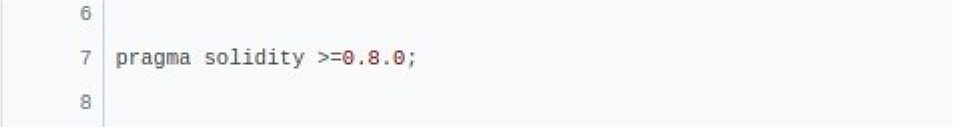
MEDIUM SEVERITY ISSUE

The audit did not find any medium severity issues.

LOW SEVERITY ISSUE

Use of floating pragma

Solidity source files indicate the versions of the compiler they can be compiled with using a pragma directive at the top of the solidity file. This can either be a floating pragma or a specific compiler version. The contract was found to be using a floating pragma which is not considered safe as it can be compiled with all the versions described.



```
pragma solidity >=0.8.0;
```

Recommended

It is always recommended that pragma should be fixed to the version.

pragma solidity 0.8.4; recommended -> compiles with 0.8.4 only