OBJECT COMPUTING

Object Computing, Inc.
12140 Woodcrest Executive Dr.
Suite 300
St. Louis, MO 63141
+1 314.579.0066

Alien Worlds Landboost Contract Audit (EOS)

Source for this analysis is github: Alien-Worlds/alienworlds-contracts
Commit tag 1a656355b7c2c217898bde31c1929ab7029fe3b6
Files contracts/landboost/landboost.[ch]pp
No ricardian contract sources were available
Business requirements and other engineering artifacts were found in the “ClickUp” document found at https://doc.clickup.com/d/h/hfd6b-9788/49593a316fe4392/hfd6b-1448

Executive Summary

No obvious signs of severe or critical security violations were detected in this audit.

Lack of ricardian contracts is not necessarily a risk but is a lost opportunity for ensuring that the coded logic is congruent with the identified business needs.

Participants

Primary Auditors were Jack DiSalvatore, with Ciju John providing a review.

Tools

Static analyzers

- EOSafe appears to be an Academic exercise last updated nearly 4 years ago
- EOSlime is a javascript tool that is also at least 2 years old
- Klevoya is a wasm analyzer tool using simple pattern matches

No static analyzers were used in this exercise. I tried to build the EOSafe tool, but it would not compile. The EOSlime tool was too complicated for me to determine how to build it. Klevoya was not useful because I could not generate an ABI file for the mining contract.

Known Vulnerabilities:
List origin github:slowmisg/**eos-smart-contract-security-best-practices**/Readme_EN.md commit tag 5f77e19e50373d341e17a003c492388e9891a2c0

- **Numerical Overflow:** when doing token math, use the `asset` object operands and do not perform operations on the `uint256` that comes from `asset.amount`
- **Authorization Check:** make sure function parameters are consistent with the caller and use `require_auth ` to check
- **Apply Check:** if the contract implements an `apply` function, bind each key action and code to meet the requirements, in order to avoid abnormal and illegal calls.
- **Transfer Error Prompt:** When processing a notification triggered by `require_recipient`, ensure that `transfer.to` is `_self`
- **Random Number Practice:** Random number generator algorithm should not introduce controllable or predictable seeds

We relied on visual inspection to look for instances of code that were similar to the examples found in the above document as well as best practices accumulated through many years of experience.

Findings And Recommendations

We examined 0 contract tables, 1 contract action, and 1 notify callback handler. The contract action is documented in clickup, but not the modifications to the private function.

- No issues

Table Signature	In ClickUp	Notes
N/A		

Action Signature	In ClickUp	Notes
ftransfer	No	No issue
withdraw	No	No issue

Test	In ClickUp	Notes
Deposit - with invalid parameter	N/A	No issue.
Deposit - wrong symbol should fail		No issue.
Deposit - with correct TLM token, should succeed		No issue
Deposit - with correct TLM token, should increase balance		No issue
Deposit - with correct TLM token, should register the deposit with the landholders contract		No issue
Withdraw - should fail without proper auth		No issue
Withdraw - should not be able to withdraw more than deposited		No issue

Withdraw - partial withdraw should succeed		No issue
Withdraw - should reduce deposit		No issue
Withdraw - full withdraw should succeed		No issue
Withdraw - should delete deposit table entry		No issue

Summary

Remark	Minor	Major	Critical
0	0	0	0

No methodology definitively proves the absence of vulnerabilities. Following assessment and remediation, modifications to an application, its platform, network environment, and new threat vectors may result in new application security vulnerabilities.