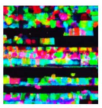

ZKSECURITY

Audit of Aleo ARC-0037

March 18th, 2024

On March 18th, 2024, zkSecurity spent two days reviewing the ARC-0037 proposal and code changes for Aleo.

The review included the official ARC document, an external design document, and implementations of the changes (mostly contained in #2385). No issues were found.

ARC-0037 introduces a way for validators (as well as their delegators) to protect against compromise of their account key. The solution introduced is to force unbonding of funds to a separate withdrawal address (potentially associated to a different key).

This withdrawal address is implemented via a new mapping in the `credits.aleo` program that mixes the logic for the Aleo token as well as the staking mechanism. Each address that has bonded funds is therefore associated with a unique withdrawal address. We illustrate the mappings of the `credits.aleo` program, including the newly added `withdraw` mapping, in the following diagram:

The withdrawal address **cannot be easily updated**, as otherwise it would allow an attacker that has compromised an account to simply update the withdrawal address to their own. Instead, the withdrawal address is set at the time of bonding, and can only be updated by first unbonding all tokens.

Thus, an attacker who manages to compromise an account must first drain the account to the previously fixed withdrawal address, which is not attacker controlled, before being able to update the withdraw address.

Even if a delegator fails to notice the update of address, in every further legitimate bonding calls they will be forced to notice a new malicious withdrawal address to the withdrawal address being required as an argument to the **`bond_public`** function:

```
finalize bond_public:
    // TRUNCATED...

    // Input the withdrawal address.
    input r2 as address.public;

    // TRUNCATED...

    // Retrieve the withdrawal address for the staker.
    get.or_use withdraw[r0] r2 into r4;
    // Ensure that the withdrawal address is consistent.
    assert.eq r2 r4;

    // TRUNCATED...

    // Set the withdrawal address.
    set r2 into withdraw[r0];
```

As such, ARC-0037 is a well-thought-out proposal that achieves its goal of de-risking the "hot" key of a staking account. The implementation is clean and well-documented, and the changes are well-contained in the **`credits.aleo`** program.