G0 Group

Security Review of

AdEx Protocol v4.1

November 22, 2019

Overview

G0 Group was engaged to perform a security review of AdEx Protocol v4.1.0 (Ethereum implementation). G0 Group was contracted for an eight person-day effort to that end. Additionally, G0 Group has previously conducted a security review of v3.2.0. The primary subjects of this review were the changes described here: namely, the introduction of the staking contract and various usability improvements to the identity contracts. This review was initially performed on https://github.com/AdExNetwork/adex-protocol-eth/tree/31cd7d27bf90a3de795be56134857382ca834951.

Files in Scope

```
contracts/
    libs/
        ChannelLibrary.sol
        MerkleProof.sol
        SafeERC20.sol
        SafeMath.sol
        SignatureValidator.sol
    AdExCore.sol
    Identity.sol
    IdentityFactory.sol
    Staking.sol
```

Result Summary

During the course of this review, 7 issues were discovered and reported. Two of these issues directly impacted security; the rest concerned usability improvements. All issues have been remediated and no further issues were discovered in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

Issues

1. The totalFunds mapping provides a misleading aggregate of the value of active bonds in a bonding pool

Type: *security /* ***Severity:*** *major*

Due to bonds staying in the pool after `willUnlock` maturation, users with mature bonds can frontrun slashing transactions: effectively inflating the pool's value in the `totalFunds` mapping without risk.

Fix Description:

Issue was fixed by removing `totalFunds` altogether (in favor of offchain accounting) and is no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

2. Overflow can lead to a bypass of input check

Type: *security /* ***Severity:*** *minor*

The require in `Staking.sol` `line 64` can be bypassed by overflow.

Fix Description:

Issue was fixed by using SafeMath, and is no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

3. Using 0x0 as the burn address makes the staking contract incompatible with a large portion of ERC20 implementations

***Type:** usability / **Severity:** major*

Many ERC20 tokens (including the current OpenZeppelin implementation) disallow transfers to the `0x0` address making them incompatible with the staking contract.

Fix Description:

Issue was fixed by changing the burn address to `0xaDbeEF0000000000000000000000000000000000` and is no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

4. Function requestUnbond in Staking.sol can be called multiple times to the detriment of the user

***Type:** usability / **Severity:** minor*

Calling `requestUnbond` multiple times extends the unbonding period each time.

Fix Description:

Issue was fixed by adding a check to `requestUnbound` which ensures it hasn’t been called yet; and no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

5. Completely slashed pools in Staking.sol destroy all new bonds upon addition

Type: *usability /* ***Severity:*** *minor*

A pool that has been completely slashed becomes a black hole: any bonds placed into it can be neither slashed nor withdrawn. A check in `addBond` to prevent posting bonds to such pools could prove useful.

Fix Description:

Issue was fixed by adding a check to `addBond` that the pool in question hasn't been maximally slashed; and is no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

6. Limiting reporting of getWithdrawAmount in Staking.sol to only msg.sender might be unnecessary.

Type: *usability /* ***Severity:*** *minor*

Fix Description:

Issue was fixed by allowing getWithdrawAmount to be called with any owner as a parameter and is no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87

7. There's no real reason to prohibit identical bonds being posted multiple times, maybe there should be an option of adding a nonce

Type: *usability /* ***Severity:*** *minor*

Fix Description:

Issue was fixed by including a nonce and no longer present in https://github.com/AdExNetwork/adex-protocol-eth/tree/e8e3d93ec61f0e7b1b3de390a407750400b37f87