NaumovLab DEVELOPMENT

NAUMOVLAB.COM
SMART-CONTRACTS.RU

NaumovLab auditors

Telegram: @radnek
Skype: mezone

Smart Contract TRONFARM

https://tronscan.org/#/contract/TKTnsntt9vz9CzHYg7dwtspK2UJqXBotSD/code
https://github.com/whalesFinance/contracts/blob/master/contracts/TRONFARM.sol

---

Произведена проверка соответствия кода байткоду:

При компиляции кода компилятором версии 0.4.25 со включенной оптимизацией полученный байткод совпадает байткодом задеплоенного смарт-контракта.

!ВАЖНО: Контракт TRONFARM взаимодействует со смарт-контрактом токена WHALES, расположенного по адресу: 0x51920f822760DD05663dd0c52FD5F3581DF673C8 (THQWi4gRDrmtCPqZArE8abuz8yjeaBZDFJ). Так же необходим аудит его кода.

Во время аудита предполагалось, что код токена соответствует коду, расположенному по адресу: https://github.com/whalesFinance/contracts/blob/master/contracts/WHALES.sol (байткод не сопоставлялся)

Во время аудита были обнаружена критические ошибки, одна из которых может быть использована как бэкдор для вывода средств пользователей админами проекта.

Описание ошибки:

1. Если пользователь желает вывести средства с контракта он вызывает метод «**leave()**»

При вызове этого метода на баланс пользователя перечисляются, ранее заблокированные, им на контракте средства и начисленные ему токены.

В случае, если баланс токенов у контракта менее необходимой для перечисления суммы токенов транзакция падает с сообщением "REVERT opcode executed. Message: SafeMath sub error". Тем самым пользователь не может вывести свои деньги с контракта до тех пор, пока контракт не будет пополнен необходимым количеством токенов.

Так как в контракте присутствует функция «**finish()**», позволяющая админам забрать баланс контракта спустя 14 дней после начала его работы, то TRX которые не смогут забрать пользователи смогут снять на свой кошелек админы.

Кроме того, манипулируя функцией «**reward()**» админы могут начислить пользователям большие суммы в токенах, что бы заблокировать возможность снятия ими средств с контракта.

2. Если пользователь внес депозит в систему, затем забрал его и повторно внес, то из-за ошибки в функции **join()**, а именно в строках:

```
if (farmer.deposited == 0) {
  farmerAddresses.push(msg.sender);
}
```

Адрес пользователя дублируется в массиве **farmerAddresses**. Из-за чего, при вызове функции **reward()**, пользователю будет начисляться двойное вознаграждение в токенах. Данную операцию пользователь может повторить многократно — увеличив свое вознаграждение в любое количество раз.

Замечания:

1. Использование цикла

for(uint i = 0; i < farmerAddresses.length; i++)

при большом количестве пользователей может привести к блокировки исполнения функции «**reward()**».

2. Начисление бонусных токенов производится с помощью вызова админами функции «**reward()**», при этом логика подразумевает, что это должно производится раз в 1 минуту. Данный способ начисления вознаграждения является плохой практикой, так как начисления бонусных токенов полностью зависит от действий админов проекта, а точность начисления — от настроек внешнего программного обеспечения и задержек в работе сети.

Заключение:

Контракт содержит критические ошибки, требующие исправления. Основная логика работы контракта реализована некорректно, рекомендуем использовать иной способ расчета бонусных вознаграждений, независящий от действий админа и времени вызова функции «**reward()**».