АУДИТ СМАРТ-КОНТРАКТА ABSOLUTUS

Huntronics.group

24.12.2019 customers@huntronics.group

Содержание

Введение

Целью проведения аудита является проанализировать и задокументировать контракты Absolutus и AbsDAO, убедиться в отсутствии известных уязвимостей, а также проверить оптимизацию контрактов. Также была поставлена задача, убедиться, что средства пользователей надежно защищены.

Методология

Аудит смарт-контрактов Absolutus и AbsDAO состоит из четырех категорий анализа.

1. **Паттерны программирования**

   Исследуется структура смарт-контракта, включающая в себя ручной и автоматический анализ. Также проверяется соответствие общепринятым практикам разработки.

2. **Статический анализ**

   Статический анализ выполняется с использованием автоматизированных инструментов, разработанных для проверки безопасности контракта.

3. **Анализ на наличие уязвимостей**

   Контракт проходит проверку для выявления распространенных уязвимостей:

   - Reentrancy. Довольно опасный недостаток, при котором контракт атакующего использует уязвимость для циклического выполнения функции в рамках одной транзакции. Такая уязвимость позволяет злоумышленнику вывести огромные суммы средств из уязвимого контракта.
   Значимость: критическая
   Не обнаружено.

   - Untrusted Delegatecall. delegatecall – инструкция, позволяющая выполнять код другого контракта, используя хранилище

отправляющего контракта. Самой простой вариацией такой атаки является уязвимость, описанная ниже.
Значимость: критическая
Не обнаружено.

- Self-Destructing, уязвимость в том, как контракт делегирует свои функции другим контрактам, которые его вызывают. Из-за такой уязвимости по ошибке был удален контракт кошелька Parity и было заморожено более 300 млн. долларов.
Значимость: критическая
Не обнаружено.

- Timestamp Dependency, уязвимость, основанная на возможности майнера манипулировать временной меткой блока в пределах 15 секунд. Данная уязвимость является критической, если в контракте используется генератор случайных чисел на основе временной метки
Значимость: средняя
Не обнаружено.

- Assertion Failure, признак того, что в потоке произошел еще один потенциально критический недостаток.
Значимость: средняя
Не обнаружено.

- Transaction-Ordering Dependence, уязвимость при которой результат выполнения транзакции меняется в зависимости от порядка выполнения транзакций в блоке.
Значимость: средняя
Не обнаружено.

4. **Ручной анализ**

Сравнение требований и реализации. Проверка смарт-контракта на соответствие указанным требованиям заказчика. Проверка на оптимизацию расхода газа и самодокументирование кода. Выполнение тестов свойств смарт-контракта.

5. Рассмотренные контракты

24 декабря, 2019 были рассмотрены следующие смарт-контракты:

- Absolutus
- AbsDAO

Версия компилятора: ^0.5.7, всего 669 строк кода.

Результат аудита

В рассмотренном контракте уязвимости не были обнаружены.

Контракт имеет достаточно хорошо спланированную структуру и оптимизированных расход газа.

Внесение средств пользователем в контракт обрабатывается в полном соответствии с логикой работы контракта. Возможности несанкционированного доступа к ним обнаружено не было. Также не было обнаружено возможности влияния владельца на контракт вне его бизнес-логики. Владелец контракта не имеет прямого доступа к средствам пользователей, лишь получает определенную комиссию.

Результаты тестов.
Пройдены тесты, полностью имитирующие работу смарт-контракта. В ходе запуска тестов проблем выявлено не было.

Обнаруженные проблемы

Проблемы перечислены от наиболее критических до наименее критических. Серьезность определяется оценкой риска эксплуатации или иного небезопасного поведения.

Уровни значимости:

- Информационный - не влияет на контракт.
- Низкий - Минимальное влияние на эксплуатационные способности.
- Средний - влияет на способность контракта действовать.
- Высокий - Влияет на способность контракта работать так, как задумано, значительным образом.
- Критический - средства могут быть распределены неправильно, потеряны или иным образом привести к значительным убыткам.

В ходе проведения аудита были выявлены мелкие недочеты низкой значимости, которые были оперативно исправлены командой разработки. Также были добавлены комментарии для упрощения понимания кода другими людьми.

Заключение

Смарт-контракты Absolutus и AbsDAO хорошо продуманы, код соответствует правилам безопасности. Проверено взаимодействие с фондом пользователя, внесение средств в него, автоматическая покупка уровней. Код написан в соответствии с Best Practices области разработки смарт-контрактов. Владелец смарт-контракта не имеет прямого доступа к средствам пользователей и возможности изменять контракт вне бизнес-логики.