HashEx

Arcona.io
Аудит безопасности смарт-контракта

Подготовлено для Arcona.io

8 апреля 2018

HashEx
https://hashex.org

Содержание

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and HashEx and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HashEx) owe no duty of care towards you or any other person, nor does HashEx make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and HashEx hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, HashEx hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HashEx, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Предпосылки

Команда Arcona.io попросила провести аудит смарт-контракта *AcronaNew.sol*. Аудит проводился с 6 апреля по 8 апреля 2018 года. Повторный аудит проводился с 11 апреля по 12 апреля 2018 года.

Аудируемый код расположен в репозитории ArconaEcosystem/arcona-system. Коммит, использованный для аудита `8ff448d3b7a8e1a008be05cc497a52c819cc71de`. Повторный аудит смарт-контракта проводился для коммита `25d1d428f838a14afe3b497ef87a045ab18a72ee`.

Целью аудита было достижение следующих целей:

- Убедиться, что смарт-контракт функционирует корректно.
- Определить потенциальные проблемы безопасности.

Информация, содержащаяся в этом отчете, должна использоваться для понимания воздействия рисков на смарт-контракт и в качестве руководства для улучшения уровня безопасности смарт-контракта путем устранения выявленных проблем.

Критические проблемы и уязвимости

Критических уязвимостей в смарт-контракте обнаружено не было.

Серьезные проблемы и уязвимости

При вызове `function changeReleaseAccount()`, если в качестве параметра `_newowner` будет передан адрес, на балансе которого имеется ненулевое значение количества токенов, то данные токены будут обнулены. При этом переменная `totalCount`, показывающая общее количество выпущенных токенов, не будет обновлена. Таким образом, инвариант равенства суммы балансов токенов на всех адресах и переменной `totalCount` не будет выполнен.

Поскольку данная функция имеет тип `internal`, проблема может возникнуть только при вызове владельцем контракта `function changeRestricted()`, `function changeRelease6m()`, `function changeRelease12m()`, `function changeRelease18m()` после проведения токенсейла и передаче в параметре `_newowner` адреса, на котором уже имеются токены. Строка 106.

Update. Исправлено. Команда Arcona.io добавила проверку на отсутствие токенов на балансе аккаунта `_newowner`: `require(balances[_newowner] == 0);` Таким образом, необходимое условие для ошибочного обнуления токенов теперь не может быть выполнено.

Незначительные проблемы

1. Установка значений аукционной продажи токенов в `function setAuction()` выполнится только в том случае, если значение параметра `_finishAuction` будет указывать на время после завершения токенсейла. Установка данного значения имеет смысл при обратном условии. `require(_finishAuction > finishSale);` необходимо заменить на `require(_finishAuction <= finishSale);` Строка 481.

   Update. Исправлено. Проверка `require(_finishAuction > finishSale);` была заменена на `require(_finishAuction <= finishSale);`

2. В whitepaper указано максимальное количество создаваемых токенов, в то же время в контракте оно не ограничено и при определенных условиях (например, при значении переменной `rateUSD` ниже 125) оно может быть превышено.
3. Рекомендуем добавить функции `increaseApproval()` и `decreaseApproval()`, позволяющие увеличить или уменьшить количество одобренных для траты токенов. В текущей реализации токена в случае если пользователь одобрил использование его токенов с помощью `function approve()` и требуется изменить количество одобренных для использования токенов, необходимо сначала установить количество одобренных токенов в 0, а затем установить новое значение. С помощью предлагаемых функций это можно будет сделать путем однократного вызова контракта.

   Update. Исправлено. Функции `increaseApproval()` и `decreaseApproval()` были добавлены.

Технические рекомендации и замечания

1. Подключение библиотеки `SafeMath` является излишним, т.к. она уже была подключена в строке 66 в контракте `BasicToken`, от которого наследуется `ArconaToken`. Строка 266.
2. Модификатор `saleIsOn` является лишним. В коде не используется и полностью по функционалу повторяет модификатор `anySaleIsOn`. Строка 319.

   Update. Исправлено. Лишний модификатор был удален.

3. В контракте указана версия Solidity не ниже 0.4.17, но контракт может быть скомпилирован только с версией не ниже 0.4.21, т.к. используется ключевое слово `emit`. Рекомендуем установить фиксированную версию 0.4.21. Строки 1, 149, 166, 245, 252.

   Update. Исправлено. Версия Solidity установлена не ниже 0.4.21.

4. Проверки `require(releaseAt(_new) == 0);` в `function changeRestricted()`, `function changeRelease6m()`, `function changeRelease12m()`, `function changeRelease18m()`, являются лишними, т.к. они производятся повторно в `function ChangeReleaseAccount()` в строке 108. Строки 357, 371, 379, 387.

   Update. Исправлено. Лишние проверки были удалены.

5. В whitepaper указано, что в течение первых 15 дней после старта продажи будет присутствовать бонус в размере 25%. В контракте же реализована другая логика: в течение первых 5 дней бонус уменьшается на 1% каждые 12 часов, в течение следующих 15 дней - на 1% каждые сутки. Строки 547-557.
6. `function checkReleaseAt()` является лишней, т.к. вызывает другую функцию `function releaseAt()` с теми же параметрами. Строка 431.

   Update. Исправлено. `function checkReleaseAt()` была удалена.

7. `function releaseAt()` является лишней, т.к. она служит для доступа к значениям `mapping(address => uint256) releaseTime;` в то время как Solidity автоматически генерирует геттеры для mapping. Строка 101.
8. `function currentPercent()` не используется и дублирует часть кода в `function createTokens()`. Рекомендуем удалить повторяющийся код из `function createTokens()` и использовать `function currentPercent()`. Строки 518-528.

   Update. Исправлено. `function currentPercent()` была удалена.

9. Модификаторы `anySaleIsOn` и `isUnderHardCap` в `function currentPercent()` являются лишними, т.к. данная функция является константной и служит для получения значения бонуса. Строка 518.

   Update. Исправлено. `function currentPercent()` была удалена.

10. Опечатка в комментарии в слове *amout*. Строка 139.

    Update. Опечатка была исправлена.

11. Функционал по хранению сертификатов и проведению токенсейла находится в коде токена. Рекомендуем вынести их в отдельные смарт-контракты для более четкого разделения функционала смарт-контрактов.
12. Переменные `isFinished` и `mintingFinished` используются одинаково. Рекомендуем провести рефакторинг, использующий только одну переменную. Строки 69, 291.
13. Для реализации функционала паузы токенсейла рекомендуем использовать контракт Pausable от Open Zeppelin.
14. В комментарии указан возврат wei, но этого не происходит в коде. Рекомендуем удалить данный комментарий. Строка 452.

Update. Исправлено. Лишний текст в комментарии был удален.

15. Переменная `mintingFinished` определена в контракте `BasicToken` и используется в модификаторе `timeAllowed` для блокировки токенов. В то же время контракт `BasicToken` не имеет отношения к эмиссии токенов. Рекомендуем провести рефакторинг для более четкого разделения функционала. Строка 69.
16. Переменные `restrictedPercent`, `refererPercent`, `first24Percent`, `auctionPercent`, `hardcap` рекомендуем объявить как константы. Строки 280-284, 296.
17. В коде используются контракты от Open Zeppelin, в них были внесены изменения. Рекомендуем провести рефакторинг путем использования оригинальных контрактов Open Zeppelin и наследования от них для добавления необходимого функционала. Данный подход позволит воспользоваться преимуществом многократного аудирования контрактов Open Zeppelin.
18. Для названий переменных с типом `mapping` (`registered`, `referral`, `certificate`) рекомендуем использовать существительные во множественном числе. Строки 276, 277, 278.
19. Название переменной `refererPercent` указывает на то, что ее значение указано в процентах, но в действительности оно указано в промилле. Рекомендуем изменить название данной переменной. Строка 281.
20. При вычислении значений переменных в токенах в качестве величины количества дробных знаков токена используется константа `18` вместо переменной `decimals`. Рекомендуем использовать переменную `decimals` для того, чтобы количество дробных знаков токена было указано только в одном месте. Строки 289-290, 298.
21. На сайте проекта в секции *AR ecosystem key features* присутствует опечатка в слове *Buld*.

    Update. Опечатка исправлена.

22. После обновления контракта в `function transferFrom()` были добавлены проверки `require(_to != address(0))` и `require(_value <= balances[_from])`. Рекомендуем добавить аналогичные проверки в `function transfer()`. Строка 145.

Выводы

В аудируемом контракте критических уязвимостей обнаружено не было. Была найдена одна серьезная проблема, при возникновении которой токены пользователя по ошибке могут быть безвозвратно обнулены. После обновления контракта проблема была ликвидирована.

Были предложены рекомендации по улучшению кода и предотвращению потенциальных атак.