Security Audit of dArt by web3go.tech

Conclusion

Audit was done by the “Web3Go” team https://web3go.tech/ (https://web3go.tech/)
by Vladimir Smelov vladimirfol@gmail.com (mailto:vladimirfol@gmail.com)
https://www.linkedin.com/in/vladimir-smelov-25021669/ (https://www.linkedin.com/in/vladimir-smelov-25021669/)

In the final contract were not found:

- Backdoors for investor funds withdrawal by anyone.
- Bugs allowing to steal money from the contract.
- Other security problems.

Obvious errors or backdoors were not found in the contract.
The client was acknowledged about all secutiry notes below.

Scope

https://github.com/dArtFlex/dArtFlex-SmartContract (https://github.com/dArtFlex/dArtFlex-SmartContract)
commit: `5c01d4aa24da6508d3398ddb693a153d0088889c`

Methodology

1. Blind audit. Try to understand the structure of the code.
2. Find info in internet.
3. Ask quiestions to developers.

4. Draw the scheme of cross-contracts interactions.
5. Write user-stories, usage cases.
6. Run static analyzers

Find problems with:

- backdoors
- bugs
- math
- potential leaking of funds
- potential locking of the contract
- validate arguments and events
- others

Result

Critical

Not found.

Major

1. Use SafeERC20

- exchange-v2/contracts/ERC20TransferProxy.sol:15
- exchange-v2/test/contracts/v2/ERC20TransferProxyTest.sol:10
- transfer-proxy/contracts/proxy/ERC20TransferProxy.sol:15

Some tokens are not fully erc20 complient (they may not return any status at all).

Recommendation.

Use SafeERC20 library.

Status.

FIXED

Warning

1. Address zero-check for attribute set.

At:

- exchange-v2/contracts/RaribleTransferManager.sol:47

- exchange-v2/contracts/RaribleTransferManager.sol:51
  and in simmilar places.
  it’s not checked that the address is not 0.
  It may be broken in front-end and lead to misusing of the contract.

Recommendation.

Add

```
require(_address != address(0), "ZERO_ADDRESS");
```

Status.

FIXED

2. Validate payouts length.

There is no validation of the input.

- exchange-v2/contracts/RaribleTransferManager.sol:165

Recommendation.

Add

```
require(payouts.length > 0, "INVALID_ARG");
```

Status.

FIXED

3. Use SafeMath or solidity ^0.8.0.

There are a lot of places where you do unsafe substraction.
It’s easy to miss a mistake.
Examples:

- exchange-v2/contracts/ExchangeV2Core.sol:61
- exchange-v2/contracts/ExchangeV2Core.sol:78
- transfer-proxy/contracts/transfer/ExchangeV2Core.sol:75
- transfer-proxy/contracts/transfer/ExchangeV2Core.sol:78
- tokens/contracts/erc-1155/ERC1155Lazy.sol:47
- exchange-v2/contracts/RaribleTransferManager.sol:107

Recommendation.

Use SafeMath everywhere or solidity ^0.8.0.

Status.

FIXED

4. Potentialy dangerous reentry.

At:

- tokens/contracts/erc-1155/ERC1155Lazy.sol:50
- exchange-v2/contracts/RoyaltiesRegistry.sol:67
- royalties-registry/contracts/RoyaltiesRegistry.sol:67
  you set the new storage set after external variable.

Recommendation.

Change state before external call.
Or/and use reentry guard.

Status.

ACKNOWLEDGED

Comment

1. Methods should be declared external.

These methods are never used internally:

```
- ExchangeV2Core.cancel (ExchangeV2Core.sol#27-32)
- ExchangeV2Core.matchOrders (ExchangeV2Core.sol#34-49)
- AssetMatcher.setAssetMatcher (AssetMatcher.sol#17-20)
- TransferExecutor.setTransferProxy (TransferExecutor.sol#28-31)
- Migrations.setCompleted (Migrations.sol#17-19)
- RoyaltiesV1Impl.getFeeRecipients (impl/RoyaltiesV1Impl.sol#10-17)
- RoyaltiesV1Impl.getFeeBps (impl/RoyaltiesV1Impl.sol#19-26)
- ERC1271.isValidSignature(bytes32,bytes) (erc-1271/ERC1271.sol#20)
- Migrations.setCompleted(uint256) (Migrations.sol#17-19)
```

Recommendation.

Make methods `external` to save gas.

Status.

FIXED

2. Use reentryGuard.

Use reentryGuard on every external/public method to be sure no reentry will happen.

Recommendation.

Use reentryGuard.

Status.

ACKNOWLEDGED

3. Function state mutability can be restricted to pure.

At:

- broken-line/contracts/LibBrokenLine.sol:74
  function state mutability can be restricted to pure.

Recommendation.

Use `pure`.

Status.

ACKNOWLEDGED

4. Comparison with boolean.

At:

- exchange-v2/contracts/RoyaltiesRegistry.sol:68
- royalties-registry/contracts/RoyaltiesRegistry.sol:68
  comparison with boolean is used.

Recommendation.

Use

```
if(!result)
```

Status.

ACKNOWLEDGED

5. Use indexed event attributes.

At:

- exchange-v2/contracts/ExchangeV2Core.sol:25
  it may be a good idea to set some attributes indexed.

Recommendation.

Set attributes indexed.

Status.

ACKNOWLEDGED

6. Resolve TODOS.

At:

- tokens/contracts/erc-1155/ERC1155Lazy.sol:97
- broken-line/contracts/LibBrokenLine.sol:16
- tokens/contracts/erc-721/ERC721Lazy.sol:52
- transfer-proxy/contracts/transfer/TransferExecutor.sol:39
  you shoould resolve TODOs.

Recommendation.

Resolve TODOs.

Status.

ACKNOWLEDGED

7. Add getRevision method.

At:

- exchange-v2/contracts/ExchangeV2Core.sol:96
  you probably already made one contract upgrade, because default `__gap` length is changed.
  You may find useful to add `getRevision` method to clearly set versions of the contract because you may misuse it without it.

Recommendation.

Add `getRevision`.

Status.

ACKNOWLEDGED

8. Potential events misordering.

At

- tokens/contracts/erc-721/ERC721Lazy.sol:57
- tokens/contracts/erc-1155/ERC1155Lazy.sol:79

Emit event before external call.
Potential mis-ordering.

Recommendation.

Be careful on frontEnd. Use noReentrant.

Status.

ACKNOWLEDGED

9. Lack of events.

You don't have much events. It will be difficult to monitor the contract activity.
e.g.:
TODO

Recommendation.

Add events with interesting argument, don't foreget about indexes.

Status.

ACKNOWLEDGED

10. Compilation-time const calculation

At

- royalties/contracts/LibRoyaltiesV1.sol:12
- royalties/contracts/LibRoyaltiesV2.sol:9
  you can just write compilation-time calculations instead of unclear hash (and then testing in constructor).

Recommendation.

Calculate const at compilation time using statement.
Somethig like:

```
bytes4 constant _INTERFACE_ID_ROYALTIES = bytes4(keccak256('getRoyalties(LibAss
```

Status.

ACKNOWLEDGED

11. Deflation tokens support.

This is not true for all tokens, that after `transfer(x)` balance will change for `x` tokens, because some contracts burn some tokens on every transfer. There were some hacks in DeFi based on this fact.

This is a valid note for all contracts which use any ERC20/BEP20 tokens.

Recommendation.

Check balances after transfer or add a comment note that such tokens are not supported.

Status.

ACKNOWLEDGED

12. Magic const.

At

- tokens-test/contracts/ERC1271.sol:8
- exchange-v2/contracts/OrderValidator.sol:16
  you have unclear magic const.

Recommendation.

Add more explanations.

Status.

ACKNOWLEDGED

13. Line slope denominator.

At

- broken-line/contracts/LibBrokenLine.sol:21
  it is not clear what is slope denomimator.

Recommendation.

Add more explanations.

Status.

ACKNOWLEDGED

14. Math optimisation.

At

- tokens/contracts/erc-1155/ERC1155Lazy.sol:47
  you can substract the value instead of setting the result of substraction.

Recommendation.

Replace

```
left = amount - transfer;
```

with

```
left -= transfer;
```

Status.

ACKNOWLEDGED

15. Use EnumerableSet.

At

- broken-line/contracts/LibBrokenLine.sol:25
- tokens/contracts/erc-1155/ERC1155Lazy.sol:73
  it is not clear why not to use EnumerableSet.

Recommendation.

Think about using the EnumerableSet.

Status.

ACKNOWLEDGED

16. Collision is possible.

At

- exchange-v2/contracts/LibOrder.sol:23
  it's ~1/4billion chance to have a collision.

Recommendation.

Think about using more bytes.

Status.

ACKNOWLEDGED

17. Revert inside the check instead of returning the status.

At

- exchange-v2/contracts/lib/LibMath.sol:21
- exchange-v2/contracts/lib/LibMath.sol:80

it make sense to revert inisde the check instead of returning the status and then reverting to use less gas.

Recommendation.

Revert inside the check.

Status.

ACKNOWLEDGED

18. Use type(uint256).max.

At

- exchange-v2/contracts/ExchangeV2Core.sol:18
  it's better to use `type(uint256).max` instead.

Recommendation.

Use `type(uint256).max` .

Status.

ACKNOWLEDGED

19. Poor comments.

In whole project is is not clear what is the purpose of the each contract and method.

Recommendation.

Add clear docstrings and comments.

Status.

ACKNOWLEDGED