Security Audit of `Octo` project contracts

Conclusion

Audit was made by Vladimir Smelov
vladimirfol@gmail.com (mailto:vladimirfol@gmail.com).

TODO In the final contract were not found:

- Backdoors for investor funds withdrawal by anyone.
- Bugs allowing to steal money from the contract.
- Any serious security problmes causing misfunctionality.

The client was acknowledged about all notes below

Scope

github.com (http://github.com):gkozyrev/studious-octo-enigma-v2
commit - `7368e313580ff3ebc2b6155e5f765cb08f25697a`

Methodology

1. Blind audit. Understand the structure of the code without reading any docs
2. Find info in internet
3. Ask questions 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

CRITICAL-1. MALICIOUS WINNER MAY BLOCK PAYOUTS.

At:

- contracts/S1CombinationToken.sol:229

if `_tokenOwner` is a contract

```
contract Hacker {
    receive() external payable {
        revert("you was hacked!");
    }
}
```

then no one will be able to get their rewards.

Recommendation.

Use push-pull, allow anyone to withdraw his reward by himself.
Why users should wait for others if they are ready to get reward?

Status.

NEW

CRITICAL-2. USING OF `_userRandomNumber` OPENS THE DOOR FOR MANIPULATIONS.

In BaseToken.sol
User can easily brute-force the random number he needs to get expected output, knowing current block.timestamp, block.number and current contract state, with for-loop oveer 100 possible numbers and dozens of possible msg.senders.
pseudo-code:

```
HackerContract
    function tryHack(state)
        for proxy in proxyList
            for i in 0 .. 100
                if randomize(BaseContractState, i, proxy) == (x, y, z, d)
                    proxy.mint(i)
    revert("not found")  # spent gas with no success
```

Recommendation.

Remove this manipulation-handle from the randomness.

Status.

NEW

CRITICAL-3. USING OF `block.timestamp`, `block.number`, `etc` OPENS THE DOOR FOR MANIPULATIONS.

- contracts/BaseToken.sol:70

this random is not safe.
It may be easily manipulated via agreement with a miner or via flashboats network.
https://www.certik.com/resources/blog/top-10-defi-security-best-practices (https://www.certik.com/resources/blog/top-10-defi-security-best-practices)
The paragraph number #3
More:
https://alcibiades.capital/blog/wolf-game-exploit/ (https://alcibiades.capital/blog/wolf-game-exploit/)
https://quantstamp.com/blog/proper-treatment-of-randomness-on-evm-compatible-networks (https://quantstamp.com/blog/proper-treatment-of-randomness-on-evm-compatible-networks)
https://arxiv.org/pdf/2206.04185.pdf (https://arxiv.org/pdf/2206.04185.pdf)
https://github.com/flashbots/mev-geth/issues/117 (https://github.com/flashbots/mev-geth/issues/117)
https://docs.flashbots.net/flashbots-auction/searchers/advanced/rpc-endpoint (https://docs.flashbots.net/flashbots-auction/searchers/advanced/rpc-endpoint)

Recommendation.

Use ChainLink VRF.
Take a look on a simmilar NFT project which uses VRF - **https://polygonscan.com/address/0x817baabb4d8e48b2eaa6f69b70f47e63867a7ee6#code** (https://polygonscan.com/address/0x817baabb4d8e48b2eaa6f69b70f47e63867a7ee6#code)
INFORMAL: This is a big problem. If you are really care about VRF price, you may accumulate requests for minting (e.g. 30 in a row) and then request VRF once for all 30 nfts. Use it if you need instant revelation.
Or you can do randomization via verifiable revelation - you publish JSON with all attributes BEFORE revelation, people just mint NFT blindly, they know only tokenId, after all nft minted, you generate ONLY ONE VRF number, then you do off-chain shuffle of the JSON based on the number (anyone

can verify it) and then you push it to IPFS as a new baseURI! So all attributes will be verifiable shuffeled at the moment of revelation!

Status.

NEW

Major

MAJOR-1. Centralization power - uncontrolled withdrawal.

At

- contracts/library/Withdrawable.sol:15
- contracts/library/Withdrawable.sol:21

it looks suspicious when owner may epmty all contract balance at any time.
INFORMAL: it’s called backdoor

Recommendation.

Introduce restrictions to decrease centralization power.

Status.

NEW

MAJOR-2. Centralization power - uncontrolled saleTax.

At

- contracts/library/ERC721Buyable.sol:55

owner may set saleTax to 100% at any time making it impossible to use NFT for any users.

Recommendation.

Introduce restrictions to decrease centralization power.
(e.g. require _tax <= 5%)

Status.

NEW

MAJOR-3. Centralization power - backdoor in S1CombinationToken.

At

- contracts/S1CombinationToken.sol:234

owner is able to flush the pool if he set `_rewards=[]`

Recommendation.

Introduce restrictions to decrease centralization power.

Status.

NEW

MAJOR-4. INCOMPLIENCY WITH EIP20 TRANSFER.

At

- contracts/library/ERC721Buyable.sol:219
- contracts/library/ERC721Buyable.sol:217
- contracts/library/ERC721Buyable.sol:166
- contracts/library/ERC721Buyable.sol:168

according to EIP20, you MUST check return success status of transfer/transferFrom.

Recommendation.

Use safeERC20.

Status.

NEW

MAJOR-5. MINT MEMBERSHIP FOR FREE.

At

- contracts/BaseToken.sol:303
- contracts/BaseToken.sol:277

if someone calls presaleMint with `_amount=0` he will get membership NFT for free.

Recommendation.

Add require amount > 0

Status.

NEW

MAJOR-6. TYPO IN IMPORTANT IDEA.

At

- contracts/BaseToken.sol:343

`_blockDifficulty` equals to the other value.

Recommendation.

Fix typo.
Also be aware that such random is not safe.

Status.

NEW

MAJOR-7. User impossible to use create 2 marketplace orders.

At

- contracts/library/ERC721Buyable.sol:118
- contracts/library/ERC721Buyable.sol:189
- contracts/library/ERC721Buyable.sol:240

what if some owner wants to sell 2-3 nft? Which nonce should it use? It's currently impossible!

Recommendation.

Use 3-levels mapping: wallet -> nftId -> nonce.

Status.

NEW

MAJOR-8. Reveal is not possible in the current implementation.

After mint all NFT parameters are public visible.
Users may go to OpenSea or another marketplace and exchange cards to build up combination.
INFORMAL: after a talk with Oleg I understood that such behaviour is not expected by you. You expect that people will get info after your revelation event.

Recommendation.

Implement true-revelation. Also check CRITICAL-3.

Status.

NEW

Warning

WARNING-1. NO DEREGISTERPROXY FUNCTION.

At

- contracts/interfaces/IWyvernProxyRegistry.sol:9

there is no declared way to de-register proxy, so it will not be able for a user to deny OpenSea later to operate over his tokens.

Recommendation.

Add deregisterProxy function.

Status.

NEW

WARNING-2. IT'S UNCLEAR HOW TO SET S1.MAXTOTALSUPPLY

At

- contracts/S1CombinationToken.sol:27

it looks that it may happen that such number of success combination will never be achievable.
You have various number of different card
(`13*4*6*6=1872`)
the same card can be minted multiple times.
There is a chance that users will never reach 380 successful combinations

Recommendation.

DISCUSS

Status.

NEW

WARNING-3. CENTRALIZATION POWER - PAUABLE MARKETPLACE.

At

- contracts/library/ERC721Buyable.sol:59

owner may set treasury to a contract which denies any eth receiving, so marketplace will be stopped.

Recommendation.

Be aware of it.
You may use push-pull-pattern.
You may ignore fails on fee sendings.

Status.

NEW

WARNING-4. WHY STORE USE ENTERED COMBINATION NAME AT BLOCKCHAIN.

At

- contracts/S1CombinationToken.sol:21

any minter can set any value to this mapping.
Even some … curse words or politician declaraions.
It's risky for marketing. Not to mention extensive gas consumption.

Recommendation.

Restrict users to set any string on blockchain.

Status.

NEW

WARNING-5. CENTRALIZATION POWER - SETBASEURI.

At

- contracts/library/Basis.sol:41
- contracts/MembershipToken.sol:116

Owner may reset all tokens metadata at any time.

Recommendation.

Let the owner to renounce the ability to reset baseURI. To change it several times and then freeze forever.

Status.

NEW

WARNING-6. CENTRALIZATION POWER - SETMAXTOTALSUPPLY.

At

- contracts/S1CombinationToken.sol:247
- contracts/BaseToken.sol:524

- contracts/BaseToken.sol:530

owner may change the rules of the game at any point of time.

Recommendation.

Restrict the owner.

Status.

NEW

WARNING-7. Be aware of OpenSea hack.

https://therecord.media/hacker-abuses-opensea-to-buy-nfts-at-older-cheaper-prices/ (https://therecord.media/hacker-abuses-opensea-to-buy-nfts-at-older-cheaper-prices/)

Recommendation.

DISCUSS

Status.

NEW

WARNING-8. Use strict equal check to the price.

At

- contracts/library/ERC721Buyable.sol:82
- contracts/BaseToken.sol:272
- contracts/BaseToken.sol:299

You use "<=" but it's safer for a user to use "==" to require him to send exact amount of price.

Recommendation.

Use strict "==" to check price.

Status.

NEW

WARNING-9. Never use payable(...).transfer().

At

- contracts/library/Withdrawable.sol:33
- contracts/library/ERC721Buyable.sol:103

- contracts/library/ERC721Buyable.sol:106
- contracts/BaseToken.sol:280
- contracts/BaseToken.sol:306
- contracts/S1CombinationToken.sol:229
- contracts/S1CombinationToken.sol:234
  you use not recommended way to send native tokens.

See - **https://consensys.net/diligence/blog/2019/09/stop-using-soliditys-transfer-now/** (https://consensys.net/diligence/blog/2019/09/stop-using-soliditys-transfer-now/)

Recommendation.

Use call with reentry protection.

Status.

NEW

WARNING-10. CONDITION MAY NEVER MET.

At

- contracts/S1CombinationToken.sol:124

You cannot force people to buy base token. It may happen that this condition will never met. But some people will want to play the game.

Recommendation.

DISCUSS

Status.

NEW

WARNING-11. PHANTOM OVERFLOW IS POSSIBLE.

At

- contracts/library/ERC721Buyable.sol:101
- contracts/library/ERC721Buyable.sol:164
- contracts/library/ERC721Buyable.sol:215

I must note that phantom overflow is possible
See - **https://medium.com/coinmonks/math-in-solidity-part-3-percents-and-proportions-4db014e080b1** (https://medium.com/coinmonks/math-in-solidity-part-3-percents-and-proportions-4db014e080b1)

Recommendation.

Be aware. Should no be the case for small decimals.

Status.

NEW

WARNING-12. Multiple membership minting to the same wallet.

At

- contracts/MembershipToken.sol:83

it may happen that minter will transfer his membership to someone else and then mint membership again.

Recommendation.

DISCUSS, is it OK?

Status.

NEW

WARNING-13. Transfer of amount=0 via erc20 sometimes impossible.

At

- contracts/library/ERC721Buyable.sol:219
  and simmilar places

I have to note that it's possible that `_price - tax` in current code could be `=0` , and some tokens dont support transfer for amount=0 so this code will fail.

Recommendation.

Check if transfer amount > 0.
Or require fee to be small.

Status.

NEW

WARNING-14. Require presale did not start.

At

- contracts/BaseToken.sol:502

it's wise to have a check that presale did not start yet.

Recommendation.

Add require.

Status.

NEW

WARNING-15. Be aware of reentry.

At

- contracts/library/ERC721Buyable.sol:103
- contracts/BaseToken.sol:277
- contracts/BaseToken.sol:280
- contracts/BaseToken.sol:303
- contracts/BaseToken.sol:306
- contracts/library/ERC721Buyable.sol:168
- contracts/library/ERC721Buyable.sol:217
- contracts/library/ERC721Buyable.sol:219
- contracts/library/ERC721Buyable.sol:86
- contracts/library/ERC721Buyable.sol:152
- contracts/library/ERC721Buyable.sol:207

you have sending eth or external contract call. Reentry may happen and influence your logic.
INFORMAL: i didnt found anything critical, but at least it may re-order your events.
also - **https://docs.soliditylang.org/en/v0.4.21/security-considerations.html#re-entrancy** (https://docs.soliditylang.org/en/v0.4.21/security-considerations.html#re-entrancy)

Recommendation.

Use nonReentrant.

Status.

NEW

Low.

LOW-1. No SPDX-License-Identifier

At

- contracts/MembershipToken.sol:1
- contracts/library/CombinableTokenBasis.sol:1

- contracts/S1CombinationToken.sol:1
- contracts/mocks/MockBaseToken4MNFT.sol:1
- contracts/interfaces/IBaseToken.sol:1
- contracts/library/Basis.sol:1
- contracts/mocks/MockSeason1CombinationToken.sol:1
- contracts/BaseToken.sol:1
- contracts/library/ERC721Buyable.sol:1
- contracts/mocks/MockERC20.sol:1

there is no `SPDX-License-Identifier` set.
At

- contracts/interfaces/IWithdrawable.sol:1
- contracts/opensea/ERC721Tradable.sol:1
- contracts/library/Withdrawable.sol:1

the `SPDX-License-Identifier` value is different from the value in others files.

Recommendation.

Use the same `SPDX-License-Identifier` value for all contracts.

Status.

NEW

LOW-2 UNCLEAR PURPOSE OF THE EMPTY INTERAFACE.

At

- contracts/interfaces/IWyvernProxyRegistry.sol:4

it’s not clear why such empty interface is needed.

Recommendation.

DISCUSS

Status.

NEW

LOW-3 USE MAPPING INSTEAD OF ARRAY TO SAVE GAS.

At

- contracts/BaseToken.sol:68

it's better to use mapping (if you dont need iteration).
Mapping is cheaper.

Recommendation.

Use mapping.

Status.

NEW

LOW-4 The check is not need.

At

- contracts/BaseToken.sol:508

this check `_presaleStartTime > 0` is not need.

Recommendation.

Status.

NEW

LOW-5 Pragma inconsistency.

At

- contracts/interfaces/IBaseToken.sol:1

the pragma solidity value is different from a value in other contract.

Recommendation.

Use the same value.

Status.

NEW

LOW-6. Index important event values.

At

- contracts/library/ERC721Buyable.sol:19-36
- contracts/S1CombinationToken.sol:38-44
- contracts/BaseToken.sol:86-87

it's wise to index important arguments to allow easy searching over blockchain logs in future.

Recommendation.

Index important event values.

Status.

NEW

LOW-7. USAGE OF MAGIC CONST.

At

- contracts/BaseToken.sol:275

magic const should be defined as a contract level constants and explained.

Recommendation.

Status.

NEW

LOW-8. EVENTS SHOULD BE PART OF INTERFACES.

At

- contracts/interfaces/ICombinableTokenBasis.sol:7
- contracts/interfaces/ICombinationToken.sol:7
- contracts/interfaces/IBaseToken.sol:5
- contracts/interfaces/IBasis.sol:7
- contracts/interfaces/IMembershipToken.sol:6

only functions are defined as a part of interface. But it's wise to also add events to the interface declaration.

Recommendation.

Events should be part of interfaces.

Status.

NEW

LOW-9. USELESS NOTICE.

At

- contracts/MembershipToken.sol:30

this comment is useless.

Recommendation.

Clear useless information from smart contracts.

Status.

NEW

LOW-10. PRETTY IMPORTS ORDER.

At

- contracts/library/Basis.sol:6

the order of imports is messed up.

Recommendation.

Reorder imports.

Status.

NEW

LOW-11. EXPLICIT VARIABLSE INITIALIZATION.

At

- contracts/library/Basis.sol:12

better to initialize it explicitly to zero.

Recommendation.

Initialize.

Status.

NEW

LOW-12. NAMING INCONSISTENCY.

At

- contracts/library/Basis.sol:16
- contracts/interfaces/IBasis.sol:7

use baseURI for consistency.

Recommendation.

use baseURI for consistency.

Status.

NEW

LOW-13. USELESS BLANK LINES.

At

- contracts/library/Basis.sol:37
- contracts/library/Basis.sol:43
- contracts/library/CombinableTokenBasis.sol:47

- contracts/library/CombinableTokenBasis.sol:53
- contracts/library/CombinableTokenBasis.sol:59
- contracts/library/ERC721Buyable.sol:61

the blank line is useless.

Recommendation.

Keep code compact.

Status.

NEW

LOW-14. DEFINE CONSTANTS AS `CONSTANT`.

At

- contracts/library/ERC721Buyable.sol:14
- contracts/BaseToken.sol:80

this should be defined as a constant.
And named in UPPER_CASE style.

Recommendation.

Define constants as `constant`.

Status.

NEW

LOW-15. UNCLEAR COMMENT.

At

- contracts/BaseToken.sol:10

it's not clear what does "using EC" means.

Recommendation.

Remove unclear comments.

Status.

NEW

LOW-16. NO CONTRACTURI IN INTERFACE.

At

- contracts/interfaces/IBasis.sol:9
- contracts/interfaces/IMembershipToken.sol:12

there is no declared view function for contractURI but only setter.

Recommendation.

Define getter as well as setter.

Status.

NEW

LOW-17. No amount in event.

At

- contracts/library/Withdrawable.sol:24

it makes sense to add amount as an event argument.

Recommendation.

Add an argument.

Status.

NEW

LOW-18. Use static sized array.

At

- contracts/S1CombinationToken.sol:19

you know for sure that the length of any array will be =4, so use static sized arrays to save gas.

Recommendation.

Use static sized array.

Status.

NEW

LOW-19. Information duplication.

At

- contracts/S1CombinationToken.sol:23
- contracts/S1CombinationToken.sol:25
- contracts/S1CombinationToken.sol:182

`baseIsCombined_[tokenId] == bool(childByParent_[tokenId ] != 0)`

Recommendation.

Get rid of duplication.

Status.

NEW

LOW-20. PURPOSELESS MODIFIERS.

At

- contracts/BaseToken.sol:212
- contracts/BaseToken.sol:231

why do you need these modifiers. This code could be part of the functions itself. You will avoid some SLOAD's.

Recommendation.

Optimize.

Status.

NEW

LOW-21. UNUSED FUNCTION ARGUMENT RAISES WARNING ON COMPILATION.

At

- contracts/library/CombinableTokenBasis.sol:65
- contracts/library/ERC721Buyable.sol:200

Recommendation.

Remove or comment out the variable name to silence this warning.

Status.

NEW

LOW-22. CHECK IS NOT NEED.

At

- contracts/library/Withdrawable.sol:29

This check is not need, because it's already checked inside of the eth transfer.

Recommendation.

Remove this check.

Status.

NEW

LOW-23. Naming typo.

At

- contracts/library/ERC721Buyable.sol:203

in fact it’s buyerSignature

Recommendation.

Rename

Status.

NEW

LOW-24. Event is not need.

At

- contracts/MembershipToken.sol:47

You can use `Transfer(from: address(0), to, tokenId)`.

- contracts/MembershipToken.sol:50

the purpose of the event is not clear.

Recommendation.

Remove this event.

Status.

NEW

LOW-25. Declare variable as immutable.

At

- contracts/opensea/ERC721Tradable.sol:10

this is immutable variable.

Recommendation.

Declare variable as immutable.

Status.

NEW

LOW-26. Unclear enums.

At

- contracts/BaseToken.sol:42

it’s not clear why all values are 2**p.
Why not use 1, 2, 3, 4 and % for random?

Recommendation.

Add any explanations as a comment.

Status.

NEW

LOW-27. Const should be UPPER_CASE.

At

- contracts/BaseToken.sol:48
- contracts/BaseToken.sol:50
- contracts/BaseToken.sol:54
- contracts/BaseToken.sol:56

Recommendation.

Const should be UPPER_CASE.

Status.

NEW

LOW-28. Use NatSpec.

At

- contracts/library/ERC721Buyable.sol:42

Recommendation.

Use NatSpec for comments for arguments.

Status.

NEW

LOW-29. Use OpenZeppelin Initializable.

At

- contracts/BaseToken.sol:62

you can take implementation of Initializable from OpenZeppelin.

Recommendation.

Use OpenZeppelin Initializable.

Status.

NEW

LOW-30. PURPOSELESS INITIALIZATION.

- contracts/BaseToken.sol:193

It is not clear why you initilize contract in 2 steps - constructor + intialize.

Recommendation.

Use constructor. Use immutable variables to save gas.

Status.

NEW

LOW-31. NO PUBLIC GETTER FOR ARRAY LENGTH.

At

- contracts/S1CombinationToken.sol:33

there is no getter for array.length.

Recommendation.

Add getter for length or for a whole array.

Status.

NEW

LOW-32. OPTIMIZE IF-CONDITION.

At

- contracts/opensea/ERC721Tradable.sol:22-26

you can use `return cond1 || cond2`

Recommendation.

Optimize.

Status.

NEW

LOW-33. UNUSED ARGUMENT.

At

- contracts/library/ERC721Buyable.sol:69

- contracts/library/ERC721Buyable.sol:140

the argument nonce is not used.

Recommendation.

Remove unused argument.

Status.

NEW

LOW-34. Use nammed args.

At

- contracts/library/ERC721Buyable.sol:74

and in a lot of places when you have more than 2 arguments it's always better to use nammed args notation to decrease the chance to misuse an argument.

Recommendation.

Use nammed args notation.

Status.

NEW

LOW-35. Fix typo.

At

- contracts/library/ERC721Buyable.sol:83

fix typo "enought".

Recommendation.

Status.

NEW

LOW-36. SLOAD optimization.

At

- contracts/BaseToken.sol:76-78

you use two timestamps always together. You can pack them in one bytes32 storage slot, to decrease SLOAD number.

Recommendation.

Status.

NEW

LOW-37. Avoid double SLOAD the same variable.

At

- contracts/BaseToken.sol:217
- contracts/BaseToken.sol:245

you have 2 SLOAD of the same variable.

Recommendation.

Optimize.

Status.

NEW

LOW-38. Identation.

At

- contracts/BaseToken.sol:246
- contracts/BaseToken.sol:349

no identation.

Recommendation.

Status.

NEW

LOW-39. Simmilar names.

At

- contracts/BaseToken.sol:370

`baseTokenMainTraits_`

and

`_baseTokenMainTraits`

are very simmilar. Easy to misuse.

Recommendation.

Rename.

Status.

NEW

LOW-40. Double condition.

At

- contracts/BaseToken.sol:462

you already checked it iin external functions.

Recommendation.

Optimize.

Status.

NEW

LOW-41. USELESS VARIABLES.

At

- contracts/S1CombinationToken.sol:112
- contracts/BaseToken.sol:213
- contracts/BaseToken.sol:214
- contracts/BaseToken.sol:232-233
- contracts/BaseToken.sol:270
- contracts/BaseToken.sol:297

msg.sender and block.timestamp does not come from structure. They comes from specific OPCALLs. You don't optimize anything.

Recommendation.

Remove them.

Status.

NEW

LOW-42. USE RELAXED COMPARISON FOR DEADLINE.

At

- contracts/library/ERC721Buyable.sol:94
- contracts/library/ERC721Buyable.sol:160
- contracts/library/ERC721Buyable.sol:211

if you want to use such mechanism inside signle transaction you may use relaxed comparison "<=" instead of "<" (as in uniswap)

Recommendation.

Use relaxed comparison for deadline.

Status.

NEW

LOW-43. ARGUMENT SHADOWS EXTERNAL DECLARATION.

At

- contracts/BaseToken.sol:113-117

Argument name shadows external declaraion.

Recommendation.

Rename.

Status.

NEW

LOW-44. OPTIMIZE CALCULATIONS.

At

- contracts/MembershipToken.sol:86
- contracts/S1CombinationToken.sol:128
- contracts/BaseToken.sol:471

you calculate the same value twice.

Recommendation.

optimize

Status.

NEW

LOW-45. OPTIMIZE TOKENURI.

At

- contracts/MembershipToken.sol:138

you can make "/" a part of the baseURI so optimize the encoding.

Recommendation.

optimize

Status.

NEW

LOW-46. USELESS `_MSGSENDER`.

At

- contracts/library/ERC721Buyable.sol:166-171
- contracts/library/ERC721Buyable.sol:220

and in all other places where you use `_msgSender()`
you never switch context, so get rid of Contex library.

Recommendation.

Use direct msg.sender etc.

Status.

NEW

LOW-47. Useless getters to internal variables.

At

- contracts/BaseToken.sol:131-184

these functions are just wasting of gas.

Recommendation.

Remove them.

Status.

NEW

LOW-48. Interface inconcictency.

At

- contracts/library/ERC721Buyable.sol:137

there is no buyer as at

- contracts/library/ERC721Buyable.sol:67

Recommendation.

Have concictent interface.

Status.

NEW

LOW-49. Discuss how is `lastTokenId_ > _rewards.length` possible.

- contracts/S1CombinationToken.sol:217

how is `lastTokenId_ > _rewards.length` possible?

Recommendation.

Discuss.

Status.

NEW

LOW-50. USE SYNTAX SUGAR FOR LIBRARIES.

At

- contracts/library/ERC721Buyable.sol:86
- contracts/library/ERC721Buyable.sol:152
-

Recommendation.

instead of

```
        require(
            SignatureChecker.isValidSignatureNow(
                _seller,
                digest,
                _sellerSignature
            ),
            "ERC721Buyable: Invalid signature"
        );
```

use syntax sugar

```
        require(
            _seller.isValidSignatureNow(
                digest,
                _sellerSignature
            ),
            "ERC721Buyable: Invalid signature"
        );
```

Status.

NEW