SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Ikonic
Date: June 27th, 2022

Document

Name	Smart Contract Code Review and Security Analysis Report for Ikonic.
Approved By	Andrew Matiukhin | CTO at Hacken OU
Type	ERC20 token; ERC721 token; ERC1155 token; NFT Marketplace
Platform	EVM
Language	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Website	https://www.Ikonic.gg
Timeline	06.05.2022 - 27.06.2022
Changelog	13.05.2022 - Initial Review 26.05.2022 - Second Review 10.06.2022 - Third Review 27.06.2022 - Fourth Review

Table of contents

Introduction

Hacken OÜ (Consultant) was contracted by Ikonic (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contracts.

Scope

The scope of the project is smart contracts in the repository:

Initial review scope

Repository:
https://github.com/ikonic-hq/Contracts

Commit:
0a69e83eb915b3cc6f33c3a2740c8e3ed496e433

Technical Documentation: No

JS tests: Yes

Contracts:

```
sale/ERC721SaleNonceHolder.sol
(sha3: 6e7374a2d80fc10037f3f677e8001986bf92cb3a52850cbe4120c430)
sale/ERC1155SaleNonceHolder.sol
(sha3: 8f67ccaa3fdc4e3f43d8fb3b1c435218a8dd1052218a79e7d92108d4)
sale/ERC721Sale.sol
(sha3: 3d2b1cca0af6a99ea46ed15be185b58d8b23e370e3eac19819840860)
sale/ERC1155Sale.sol
(sha3: 9c85cd14ac70b68754f3a022f6a0a6112f2c6d50fdec51ae3c4156c5)
proxy/TransferProxy.sol
(sha3: b0873901da0ff150b91c42a9fe262ad2a26551a964b88d739cb74de1)
proxy/ServiceFeeProxy.sol
(sha3: d189e18633373fa50bf440660372c5941e3cf42dadd7c6597c1f1425)
roles/OperatorRole.sol
(sha3: a0e4a40582774a3337cb63afd46f63ea1dce887f534342529e61f707)
libs/UintLibrary.sol
(sha3: 1e7a76c3365c550fc2c1169029f6d763a153a3bc07875e9b365125d7)
libs/StringLibrary.sol
(sha3: 38c55c5f2d465dcec3c1155dece9a9a92d0e5d6fe41c008abf3bd6dd)
tokens/ERC1155Base.sol
(sha3: 06a79d12cdba762371e56799487709f42d5569cde0fd89155fa72252)
tokens/ERC721Base.sol
(sha3: b1e03b552f88dd42a14ffa6b8bed217fb8d4d2a1c01ab38484c64061)
tokens/IkonicERC721Token.sol
(sha3: 054bd02dbe9cbaa38fcde05b61234b29419e1e51c10548f0403dc7b7)
tokens/HasSecondarySale.sol
(sha3: 465a259f3f67b8865d7c30a0d7194a8746deb8e4a6dd7f7e7f44de91)
tokens/ERC2981PerTokenRoyalties.sol
(sha3: 8b854030e8033d2caac4019ebf21ef6671c12b170f77da52b26352bf)
tokens/HasTokenURI.sol
(sha3: 8069e951ac8d396836ec1dc210f60b7cc87c45fa866e281c022af62c)
tokens/IkonicERC1155Token.sol
(sha3: eff509b36c3d4e6839774dbebdd61f060610ef20ab612115a01eb2c2)
tokens/IkonicToken.sol
(sha3: 00339e04c28c45548ba50af338b73fe5d9a9c6bc6f77acb74b01d1fb)
```

```
        tokens/ERC2981Base.sol
        (sha3: 4ef865b36cea4f92d633def16489864afbe96c4b75680e9469307ee9)
        service_fee/ServiceFee.sol
        (sha3: 79b61f5666e70de46dfa5291ab163af577c958726623015bd3984040)
```

Second review scope

Repository:
https://github.com/ikonic-hq/Contracts
Commit:
1c90d0c0d173539960a840148289e3f6b0a0bebe
Technical Documentation: No
JS tests: Yes
Contracts:

```
        sale/ERC721SaleNonceHolder.sol
        (sha3: 6e7374a2d80fc10037f3f677e8001986bf92cb3a52850cbe4120c430)
        sale/ERC1155SaleNonceHolder.sol
        (sha3: 8f67ccaa3fdc4e3f43d8fb3b1c435218a8dd1052218a79e7d92108d4)
        sale/ERC721Sale.sol
        (sha3: 72cb8f389ddec0334833b5dd5eb18890dbdaeef72a68aead93028285)
        sale/ERC1155Sale.sol
        (sha3: 3287f43c4c626e3833e7fa746a596ada3e3ead0a3db9d6e32b517c42)
        proxy/TransferProxy.sol
        (sha3: b0873901da0ff150b91c42a9fe262ad2a26551a964b88d739cb74de1)
        proxy/ServiceFeeProxy.sol
        (sha3: d189e18633373fa50bf440660372c5941e3cf42dadd7c6597c1f1425)
        roles/OperatorRole.sol
        (sha3: f5b7a86d4493e754b19801fc4e104c5c9dced0dd29fbc18ac974e897)
        libs/UintLibrary.sol
        (sha3: 1e7a76c3365c550fc2c1169029f6d763a153a3bc07875e9b365125d7)
        libs/StringLibrary.sol
        (sha3: 38c55c5f2d465dcec3c1155dece9a9a92d0e5d6fe41c008abf3bd6dd)
        tokens/ERC1155Base.sol
        (sha3: 1dad2f18cf15489db38f5ee99cfe25454ca2ed6922065979afa881b9)
        tokens/ERC721Base.sol
        (sha3: b1e03b552f88dd42a14ffa6b8bed217fb8d4d2a1c01ab38484c64061)
        tokens/IkonicERC721Token.sol
        (sha3: 7ddd360d0a1f7e68f96e9624c6fb30e4e7a4cd0c2deeb280aec686f6)
        tokens/HasSecondarySale.sol
        (sha3: 465a259f3f67b8865d7c30a0d7194a8746deb8e4a6dd7f7e7f44de91)
        tokens/ERC2981PerTokenRoyalties.sol
        (sha3: 8b854030e8033d2caac4019ebf21ef6671c12b170f77da52b26352bf)
        tokens/HasTokenURI.sol
        (sha3: 8069e951ac8d396836ec1dc210f60b7cc87c45fa866e281c022af62c)
        tokens/IkonicERC1155Token.sol
        (sha3: 7a69f63cfeb0d87d8cc02c7fd586d4bcbd5403fe016031c44ea4ce8c)
        tokens/IkonicToken.sol
        (sha3: 080a451c69b02483e1411d1156a62d41a51274fd2355f5737d4ad7f9)
        tokens/ERC2981Base.sol
        (sha3: 4ef865b36cea4f92d633def16489864afbe96c4b75680e9469307ee9)
        service_fee/ServiceFee.sol
        (sha3: 9b8b9f119ebfe4dee99e2ec66f72f6544dcfaec8c128722fd7980cf9)
        interfaces/IIkonicERC1155Token.sol
        (sha3: 68578e8a2fda68f790f83904dc546119f45f0f622a5d1eed8572e3e6)
        interfaces/IIkonicERC721Token.sol
        (sha3: 5e31d2897835cc16420e4f26968307d0fcc055828825aa04b680d7ca)
        interfaces/IIkonicToken.sol
```

(sha3: b10105dc062fca4b1e3500dc0615cfa81a9170cf0c1f56d3ffb15997)
interfaces/IERC2981Royalties.sol
(sha3: 58b5d1beeacffbad87e7b0ebc9edb48205bf022560954551b31809b6)
interfaces/IServiceFee.sol
(sha3: a06e3c1bd7b273fe79558feb74ab0d6ccb2b140c641ad54fb4a54c7a)

Third review scope

Repository:
https://github.com/ikonic-hq/Contracts
Commit:
e46735ed2b8be09b3cec365cc24a4877b2e08bca
Technical Documentation: No
JS tests: Yes
Contracts:
sale/ERC721SaleNonceHolder.sol
(sha3: 3ec37720699167de995f378e31def154f143e39eee19595d7b4f9f3d)
sale/ERC1155SaleNonceHolder.sol
(sha3: 36370f72dabc6c12ce314ecd0516300dfa63f10123ee3228e932856e)
sale/ERC721Sale.sol
(sha3: 0c4765b794abf7e9fa8d9671a8d6d3abdc6a61af4ea95f1f9e29b8b7)
sale/ERC1155Sale.sol
(sha3: 18c24cca41b7e5de412cfe6ecb2336a4ff3d5f500053aa312c9c876f)
proxy/TransferProxy.sol
(sha3: d1f166929271dc1dc59d717116999f802e7be6e1ed6cd52519002a3c)
proxy/ServiceFeeProxy.sol
(sha3: 6bac410aea087ccb7982ff9e31d3445e47727d117b18aad2fc470416)
roles/OperatorRole.sol
(sha3: bfacbca8d76dc1142787734b254dbfe4567929a0122f871d0446b734)
tokens/ERC1155Base.sol
(sha3: 1e920e9a255660bd8e1bfc786fb67d226afbd961e0d112ea2e9949d4)
tokens/ERC721Base.sol
(sha3: c039d50a6effd2206cefb793f83f5332b47ce8798653fc1e12f10647)
tokens/IkonicERC721Token.sol
(sha3: b4691f46cb3a1df791337d096e282e7fb6c3e710e074a15cf1c270fa)
tokens/HasSecondarySale.sol
(sha3: 0cb6ae95993757537457109b5a5c3b4764de056b26dc99f477d1a1ca)
tokens/ERC2981PerTokenRoyalties.sol
(sha3: 50342f3ec38739e42c1a961ce6d9257985b2f5112a7b0aee6d11d95d)
tokens/HasTokenURI.sol
(sha3: e7fc22e15bce161777d9aa036b9064f3887135a47408d39f77e52498)
tokens/IkonicERC1155Token.sol
(sha3: 961ef0a0e4011c4bec1e154df316c56c9618d4ecb11602088b6c9f72)
tokens/ERC2981Base.sol
(sha3: 62ab726864768b4c6b2f00f9e07253d8d8b5e7bbcc5abcceec10610a)
service_fee/ServiceFee.sol
(sha3: 5986c6d4be15809594c1920346eb40b3b65201485ad597c7364c10ea)

Fourth review scope

Repository:
https://github.com/ikonic-hq/Contracts
Commit:
f66e8f94055608f79da10e13d6861870c6689442
Technical Documentation: Yes
- https://docs.google.com/document/d/1Wkhfsgh-itDUihONak8bCbKKX2ZfijjlFOIh4GE436A/edit
- https://drive.google.com/file/d/1xqtMl54h3iB3qW76ahY6FuDBJXc5hqaY/view

JS tests: Yes
Contracts:

```
      sale/ERC721SaleNonceHolder.sol
      (sha3: 020b79df0ee05a3176dfb053ccb0c1d07f74f8651ef5a575395b0cc2)
      sale/ERC1155SaleNonceHolder.sol
      (sha3: 209787ffab3bf1d21acda3d4e4d58ea43a677d89924fa4630fcc5423)
      sale/ERC721Sale.sol
      (sha3: b8f43b16c46a093d76911deeb05aba3f334fc01efc81cabc6cf2f407)
      sale/ERC1155Sale.sol
      (sha3: 85f29c3b1ff8956bb2d1096182204255aefcdc41b4fcc9e456dcb94f)
      proxy/TransferProxy.sol
      (sha3: f3ffecfd9f56a7af811ae058da03ebc693b7351c36e895c4d3c96b7c)
      proxy/ServiceFeeProxy.sol
      (sha3: feaf926cb7c4c8302bd9fc6cf48e2a1ea431d1839fee8bb70a0b099f)
      Migrations.sol
      (sha3: ba173ff418de0f63d81110e058cf097a4a7db202dd78991fe5b1a49f)
      roles/OperatorRole.sol
      (sha3: 47f25ad0ff0fca6b71dc6a25aa265d203b782d44a88d5666776e95e6)
      libs/UintLibrary.sol
      (sha3: b71b41e3a7afa8f0002420a56dc0729c390e6881518d62ee22301ad8)
      libs/StringLibrary.sol
      (sha3: fe8811871bd692a70fbf22308c4c0b689470e7cf1a3d1bd994eb5a05)
      tokens/ERC1155Base.sol
      (sha3: 7b1edebcf1cc489a216b0bc8a4ba7987b4c0b07fe622fdea99332c64)
      tokens/ERC721Base.sol
      (sha3: 1c27261056ae634eacfa1f4091d075f0fb9f0fd16329cd93c998a0a8)
      tokens/IkonicERC721Token.sol
      (sha3: ef284a81a35375bd0dbea22b0a0a873f7e71874306d1057cc8cf27e5)
      tokens/HasSecondarySale.sol
      (sha3: 43bbefa814db545b249c1e3a331f351fa9c4010dc7135ad1385a81d7)
      tokens/ERC2981PerTokenRoyalties.sol
      (sha3: 243b78e053f5d4aa4f12bc2b153d430e12fdacb9cf2e401fce21c342)
      tokens/HasTokenURI.sol
      (sha3: 01380c296c6d3c96813aae6f66579baec3a9eeea6a9020b13f5328e3)
      tokens/IkonicERC1155Token.sol
      (sha3: 592c4687c3a30f0094e7ad8b13bce520c04ade9e09ca765d5043c773)
      tokens/IkonicToken.sol
      (sha3: 84432d8f5a04ab0d64ce03ab9805d9a647bb56d675dc556b50530988)
      tokens/ERC2981Base.sol
      (sha3: ad54228c461a15f5f568819ef36c0abf911f4a09a948387d7cb0cb7c)
      service_fee/ServiceFee.sol
      (sha3: dc162d56e084fafb8b83d30c62d33c4de115613964c222ddb58b3a42)
```

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they cannot lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that cannot have a significant impact on execution

Executive Summary

The score measurement details can be found in the corresponding section of the methodology.

Documentation quality

The Customer provided no functional requirements and some technical specifications. The total Documentation Quality score is **6** out of **10**.

Code quality

The total CodeQuality score is **9** out of **10**. Code duplications. Good NatSpecs.

Architecture quality

The architecture quality score is **6** out of **10**. The overall architecture is quite clean. Functions are overwhelmed with template code that could be moved to separate functions and be reused.

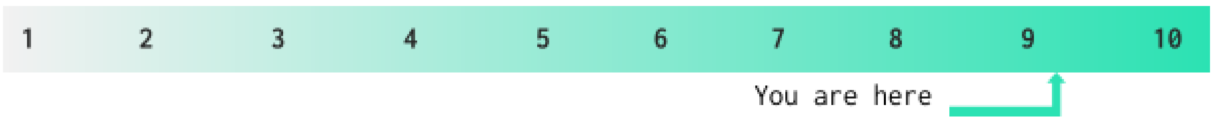
Security score

As a result of the audit, security engineers found **no security issues**. The security score is **10** out of **10**.

All found issues are displayed in the “Findings” section.

Summary

According to the assessment, the Customer's smart contract has the following score: **9.1**



Findings

■■■■ Critical

No critical severity issues were found.

■■■ High

Tests failing

42 tests out of a total of 61 provided are failing. Failing tests may point to either incorrect business logic implementation or some mistakes in the tests themselves.

Scope: tests

Recommendation: ensure tests run smoothly and cover at least 95-100% of code lines and all logic branches.

Status: Fixed (Revised Commit: f66e8f9)

■■ Medium

1. **Tautology or contradiction**

   It is tautological to check if the unsigned integer variable is greater or equal to zero. While it is unsigned, it could not be less than zero.

   Contracts: ERC721Sale, ERC1155Sale

   Functions: sell, updatePriceAndCurrency

   Recommendation: fix the incorrect comparison by changing the value type or the comparison.

   Status: Fixed (Commit Hash: 1c90d0c)

2. **Contract inconsistencies**

   While the `cancel` function checks the owner of the initiated sale, the `updatePriceAndCurrency` and `updateExpSaleDate` functions check the actual owner of the token. In all cases, the error message, when the check is not fulfilled, says: “Caller is not the owner of the token”.

   Contracts: ERC721Sale

   Functions: cancel, updatePriceAndCurrency, updateExpSaleDate

   Recommendation: revise the logic and messages.

   Status: Fixed (Commit Hash: 1c90d0c)

■ Low

1. **Floating solidity version**

   It is recommended to specify the exact Solidity version in the contracts.

Recommendation: specify the exact Solidity version (ex. pragma solidity 0.8.4 instead of pragma solidity ^0.8.4).

Status: Fixed (Revised Commit: e46735e)

2. **Using experimental ABIEncoderV2 pragma.**

   Starting Solidity version 0.8.0 the ABI coder v2 is activated by default. Choose the old behavior using `**pragma abicoder v1;**`. The pragma `**pragma experimental ABIEncoderV2;**` is still valid, but it is **deprecated** and has no effect.

   Contracts: IServiceFee, ServiceFee, OperatorRole, ERC1155Sale, ERC721Sale, IkonicERC1155Token, IkonicERC721Token

   Recommendation: remove the deprecated pragma. To be explicit, use `pragma abicoder v2`; instead.

   Status: Fixed (Commit Hash: 1c90d0c)

3. **Tautology with booleans**

   It is sufficient to return the logical value itself instead of returning true in the positive branch and false in the negative branch in the if-operator.

   Contracts: ERC721Sale, ERC1155Sale

   Functions: isCurrencyValid

   Recommendation: return boolean statement instead

   Status: Fixed (Commit Hash: 1c90d0c)

4. **Excess state reading**

   The value of the `saleInfos[address(_token)][_tokenId].owner` is being read twice: on the line 194 and 204.

   Contracts: ERC721Sale

   Functions: buy

   Recommendation: move the `saleOwner` assignment to the topmost position in the function and use its value in the `require` statement.

   Status: Fixed (Commit Hash: 1c90d0c)

5. **Reading the state in the loop**

   It is not recommended to read the state variable in the loop because of burning Gas.

   Contracts: ERC721Sale, ERC1155Sale

   Functions: addSupportCurrency

   Recommendation: assign `supportCurrencyName.length` to a local memory variable and use it then in the loop.

Status: Fixed (Commit Hash: 1c90d0c)

6. **Excess external calling**

The function call `_token.balanceOf(msg.sender, _tokenId)` is being done twice: on the lines 158 and 163.

Contracts: ERC1155Sale

Functions: sell

Recommendation: store the result of the first call to a local memory variable and then use it in both `require` statements.

Status: Fixed (Commit Hash: 1c90d0c)

7. **A state variable could be immutable**

A state variable that never changes its value and is initialized in the constructor should be declared **immutable** to save Gas.

Contracts: IkonicToken

Variable: _totalSupply

Recommendation: use the keyword `immutable` to declare the state variable immutable.

Status: Fixed (Commit Hash: 1c90d0c)

8. **A state variable could be constant**

A state variable that never changes its value should be declared **constant** to save Gas.

Contracts: IkonicToken

Variables: _decimals, _name, _symbol

Recommendation: use the keyword `constant` for state variables that never change their values.

Status: Fixed (Commit Hash: 1c90d0c)

9. **Boolean equality**

Boolean constants can be used directly and do not need to be compared to **true** or **false**.

Contract: ERC1155Base

Function: burn

Recommendation: remove the equality to the boolean constant.

Status: Fixed (Commit Hash: 1c90d0c)

10. **Implicit variables visibility**

State variables (or constants) that do not have specified visibility are declared **internal** implicitly. That could be not obvious.

Contract: IkonicToken.sol

Constants: _totalSupply, _decimals, _name, _symbol

Recommendation: always declare visibility explicitly.

Status: Fixed (Revised Commit: e46735e)

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit cannot guarantee the explicit security of the audited smart contracts.