PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For YetiSwap NFT Marketplace

02 September 2021

paladinsec.co

info@paladinsec.co

Table of Contents

Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

1 Overview

This report has been prepared for YetiSwap's NFT Marketplace contract on the Avalanche network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	YetiSwap NFT Marketplace
URL	https://www.yetiswap.app/
Platform	Avalanche
Language	Solidity

1.2 Contracts Assessed

Name	Contract	Live Code Match
`YetiSwapNFTMarketV1`	`0x14390F57CCFdB45f969381e7e107AcF062d3A592`	MATCH

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High	1	1	-	-
Medium	1	1	-	-
Low	1	-	-	1
Informational	8	7	-	1
Total	**11**	**9**	**-**	**2**

Classification of Issues

Severity	Description
High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
Informational	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 YetiSwapNFTMarketV1

ID	Severity	Summary	Status
01	HIGH	Lack of validation on commission rates allows the marketplace owner to claim all NFTs for sale	RESOLVED
02	MEDIUM	Purchases will revert if `comissionTaker` is set to the zero address	RESOLVED
03	LOW	Sales could be denied if `comissionTaker` or the NFT creator are non-payable contracts	ACKNOWLEDGED
04	INFO	Typographical errors in error messages and variables	RESOLVED
05	INFO	It is non-trivial for users to find out which tokens they have on sale	ACKNOWLEDGED
06	INFO	The `yetiswapNFT` should not be addable to the whitelist	RESOLVED
07	INFO	Requiring a non-negative number to be greater or equal to zero is futile	RESOLVED
08	INFO	Inconsistent usage of SafeMath in incrementing	RESOLVED
09	INFO	Lack of events for `setComissionRate`, `setComissionRateAvax`, `setComissionTaker`, `addNewCategory`, `addWhitelistedCollection`, `removeWhitlistedCollection`, `addWhitelistedToken`, `removeWhitelistedToken` and `updateCreatorRate`	RESOLVED
10	INFO	`getWhitelistedCollection`, `getWhitelistedToken`, `addWhitelistedCollection`, `removeWhitelistedCollection`, `addWhitelistedToken`, `removeWhitelistedToken` and `updateCreatorRate` can be made `external`	RESOLVED
11	INFO	`baseToken` and `yetiswapNFT` can be made `immutable`	RESOLVED

2 Findings

2.1 YetiSwapNFTMarketV1

The YetiSwapNFTMarketV1 is a simple NFT marketplace where users can sell NFTs (ERC721 tokens) for a list of ERC-20 tokens or AVAX. With each sale, a commission is distributed to both the creator of the NFT (configurable by the admin) and the YetiSwap team themselves. The latter commission when the NFT is listed for sale is in the native YetiSwap currency.

Only whitelisted NFT collections can be sold and only whitelisted currencies can be accepted.

❗ **Disclaimer:** The contract contains references to the yetiswapNFT token which was not included in this audit.

2.1.1 Privileged Roles

The following functions can be called by the owner of the contract:

- `setComissionRate`
- `setComissionRateAvax`
- `setComissionTaker`
- `addNewCategory`
- `addWhitelistedCollection`
- `removeWhitelistedCollection`
- `addWhitelistedToken`
- `removeWhitelistedToken`
- `updateCreatorRate`

2.1.2 Issues & Recommendations

Issue #01	**Lack of validation on commission rates allows the marketplace owner to claim all NFTs for sale**
Severity	HIGH SEVERITY
Description	Currently the `setComissionRate`, `setComissionRateAvax` and `updateCreatorRate` functions have no upper limit. This allows the marketplace owner to set these variables to 100% to potentially purchase all NFTs and receive a full refund afterwards. This privilege might scare away users into listing their NFTs.
Recommendation	Consider adding reasonable limits to the commission rates. For example a summed maximum (all rates together) of 30% could be considered.
Resolution	RESOLVED All fees can be individually set to at most 15%, resulting in a total maximum fee of 30%.

Issue #02	**Purchases will revert if `comissionTaker` is set to the zero address**
Severity	MEDIUM SEVERITY
Description	The marketplace owner can designate a wallet to receive the commission fees. If this wallet is set to the zero address, most transfers will however revert since ERC-20 tokens often prevent token transfers to the zero address.
Recommendation	Consider adding a check to ensure that the `_commissionTaker` is not equal to zero when it is updated. `require(_commissionTaker != address(0));`
Resolution	RESOLVED The recommendation has been added.

Issue #03	**Sales could be denied if `comissionTaker` or the NFT creator are non-payable contracts**
Severity	LOW SEVERITY
Description	In case the `comissionTaker` or NFT creator is a contract, AVAX sales might be blocked as AVAX transfers can revert to their address. This might not be desirable behavior.
Recommendation	Consider changing the `transfer()` calls to `call()` calls as is always recommended. Furthermore, consider checking if the call was successful using the success `return` variable, and if it is not, the AVAX could be sent to the buyer instead of reverting the call to ensure purchases are always possible.
Resolution	ACKNOWLEDGED

Issue #04	Typographical errors in error messages and variables
Severity	INFORMATIONAL
Description	The contract contains an array of spelling errors. This might not look good when users receive an error with a typographical error.
Recommendation	Consider resolving the following spelling errors.

Original	Suggestion
comission	commission
cant	can't
This sale not active	This sale is no longer active
No approve for nft first you need to give approve to this contract!	Contract not approved to take NFT. Approve the contract first.
Not active category!	Category not active
_price amount must bigger than 0	price cannot be zero
You must be seller of this NFT	You must be the seller of this NFT
Nft must be sellign with ERC20	NFT must be bought with AVAX
Seller amount must bigger than 0	Seller amount must be bigger than zero
Message value must be equal price	Insufficient AVAX received
differnt	different
_categorieName	_categoryName

Resolution	RESOLVED

Issue #05	It is non-trivial for users to find out which tokens they have on sale
Severity	INFORMATIONAL
Description	Currently there are no functions to get the active sales or sales history for a specific user – this thus needs to be properly indexed on the frontend.
Recommendation	Consider first of all adding `indexed` to the `seller`, `buyer`, `saleIndex`, `nftContract` and `nftId` variables in the events. This allows users, tools and your frontend to efficiently filter the events with web3. Secondly, consider adding a list for each user containing the indices of their (active) listing history. Finally, it might be useful to add a timestamp variable to the `SaleNFT` struct that indicates when the sale was created.
Resolution	ACKNOWLEDGED The client has indicated that they have an off-chain graph database that takes care of this.

Issue #06	The `yetiswapNFT` should not be addable to the whitelist
Severity	INFORMATIONAL
Description	We see no value in being able to add the `yetiswapNFT` token to the whitelisted contracts since it has unique business logic for all use cases.
Recommendation	Consider whether `yetiswapNFT` ever has to be whitelisted, and if not, consider preventing this possibility to reduce the risk of misconfiguration. `require(collectionContract != yetiswapNFT);`
Resolution	RESOLVED The recommendation has been implemented.

Issue #07	Requiring a non-negative number to be greater or equal to zero is futile
Severity	INFORMATIONAL

Location	Line 238
	`require( _comissionRateYTS >= 0,"Comission must be equal or bigger than 0");`
	
	Line 244
	`require( _comissionRateNonYTS >= 0,"Comission must be equal or bigger than 0");`

Description	These two lines check that an unsigned integer is greater than or equal to zero. However, this will always be the case so these checks are redundant.
Recommendation	Consider removing these redundant requirements.
Resolution	RESOLVED

Issue #08	Inconsistent usage of SafeMath in incrementing
Severity	INFORMATIONAL

Location	Line 150
	`onSaleNftAmount++;`
	
	Line 271
	`numCollections = numCollections.add(1);`

Description	Throughout the contract, both unsafe and safe addition and subtraction is used. Although we could not find any problem with either option since it is unlikely for these variables to overflow or underflow, a consistent codebase can look more appealing to investors and third-party reviewers.
Recommendation	Consider committing to a single notation for both increments (`.add`) and decrements (`.sub`).
Resolution	RESOLVED

Issue #09	**Lack of events for `setComissionRate`, `setComissionRateAvax`, `setComissionTaker`, `addNewCategory`, `addWhitelistedCollection`, `removeWhitlistedCollection`, `addWhitelistedToken`, `removeWhitelistedToken` and `updateCreatorRate`**
Severity	INFORMATIONAL
Description	Functions that affect the status of sensitive variables should emit events as notifications.
Recommendation	Add events to the above functions.
Resolution	RESOLVED

Issue #10	**`getWhitelistedCollection`, `getWhitelistedToken`, `addWhitelistedCollection`, `removeWhitelistedCollection`, `addWhitelistedToken`, `removeWhitelistedToken` and `updateCreatorRate` can be made `external`**
Severity	INFORMATIONAL
Description	Functions not used internally in a contrct can be changed from `public` to `external`. Apart from being a best practice when the function is not used within the contract, this can lead to a lower gas usage in certain cases.
Recommendation	Consider making these functions `external`.
Resolution	RESOLVED

Issue #11	**`baseToken` and `yetiswapNFT` can be made `immutable`**
Severity	INFORMATIONAL
Description	Variables that are only set in the constructor but never modified can be indicated as such with the `immutable` keyword. This is considered best practice since it makes the code more accessible for third-party reviewers and saves gas.
Recommendation	Consider making `baseToken` and `yetiswapNFT` explicitly `immutable`.
Resolution	RESOLVED


PALADIN
BLOCKCHAIN SECURITY