Smart Contract

Security Assessment

For LifeStyleDAO Marketplace
02 Feb 2023

Ascendant

Ascendant
@ascendantproj
www.ascendant.finance

Ascendant

Table of Contents

3	Disclaimer
4	Executive Summary
5	Overview
6	Findings Summary & Legend
9	Manual Review • Issue Checking Status • Audit Findings • Functional Test Status
21	Automated Review • Unified Model Language
23	Conclusion

DISCLAIMER

Ascendant Finance ("Ascendant") 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 Ascendant.

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 Ascendant is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Ascendant 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.

Ascendant retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Ascendant is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. Ascendant may, at its discretion, claim bug bounties from third-parties while doing so.

Executive Summary

Severity	Found
High	1
Medium	1
Low	29
Informational	78
Total	109

We performed an independent technical audit to identify Smart Contracts uncertainties. This shall protect the code from illegitimate authorization attempts or external & internal threats of any type. This also ensures end-to-end proofing of the contract from frauds. The audit was performed semi-manually. We analyzed the Smart Contracts code line-by-line and used an automation tool to report any suspicious code.

The following tools were used:

- Truffle
- Remix IDE
- Slither

Overview

This report has been prepared for LifestyleDAO Marketplace on the Ethereum network. Ascendant provides a user-centered examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

Summary

Project Name	LifestyleDAOMarketplace
Platform	Ethereum
Language	Solidity

Contracts Assessed

Name	Location
DAOMarketplace.sol	Goerli: 0x9De96D05A79c3CBf97b13Ecf424B39090932FE91
EnumerableSetUpgradeable.sol	In DAOMarketplace contract
MathUpgradeable.sol	In DAOMarketplace contract
ContextUpgradeable.sol	In DAOMarketplace contract
AddressUpgradeable.sol	In DAOMarketplace contract
IERC721ReceiverUpgradeable.sol	In DAOMarketplace contract
IERC1155ReceiverUpgradeable.sol	In DAOMarketplace contract

Name	Location
IERC1155Upgradeable.sol	In DAOMarketplace contract
IERC165Upgradeable.sol	In DAOMarketplace contract
IERC20Upgradeable.sol	In DAOMarketplace contract
IERC20.sol	In DAOMarketplace contract
ERC165Upgradeable.sol	In DAOMarketplace contract
IERC721Upgradeable.sol	In DAOMarketplace contract
StringsUpgradeable.sol	In DAOMarketplace contract
Intializable.sol	In DAOMarketplace contract
MulticallUpgradeable.sol	In DAOMarketplace contract
ReentrancyGuardUpgradeable.sol	In DAOMarketplace contract
IAccessControlUpgradeable.sol	In DAOMarketplace contract
IAccessControlEnumerableUpgradeable.sol	In DAOMarketplace contract
AccessControlUpgradeable.sol	In DAOMarketplace contract
AccessControlEnumerableUpgradeable.sol	In DAOMarketplace contract
IERC2981Upgradeable.sol	In DAOMarketplace contract
TWAddress.sol	In DAOMarketplace contract
SafeERC20.sol	In DAOMarketplace contract
IWETH.sol	In DAOMarketplace contract
WETH.sol	In DAOMarketplace contract

Name	Location
CurrencyTransferLib.sol	In DAOMarketplace contract
ERC2771Upgradeable.sol	In DAOMarketplace contract
IPlatformFee.sol	In DAOMarketplace contract
IThirdwebContract.sol	In DAOMarketplace contract
ERC165Upgradeable.sol	In DAOMarketplace contract
StringsUpgradeable.sol	In DAOMarketplace contract
Intializable.sol	In DAOMarketplace contract
MulticallUpgradeable.sol	In DAOMarketplace contract
ReentrancyGuardUpgradeable.sol	In DAOMarketplace contract
IAccessControlUpgradeable.sol	In DAOMarketplace contract
IAccessControlEnumerableUpgradeable.sol	In DAOMarketplace contract
AccessControlUpgradeable.sol	In DAOMarketplace contract
AccessControlEnumerableUpgradeable.sol	In DAOMarketplace contract
IERC2981Upgradeable.sol	In DAOMarketplace contract
TWAddress.sol	In DAOMarketplace contract
SafeERC20.sol	In DAOMarketplace contract
IWETH.sol	In DAOMarketplace contract
WETH.sol	In DAOMarketplace contract

Findings Summary

Severity	Found
High	1
Medium	1
Low	29
Informational	78
Total	109

Classification of Issues

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 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.

Manual Review

Issues Checking Status

Issue Description	Checking Status
Compiler errors	PASS
Race conditions and Reentrancy. Cross-function race conditions.	PASS
Possible delays in data delivery.	PASS
Oracle calls.	PASS
Front running.	PASS
Timestamp dependence.	PASS
Integer Overflow and Underflow.	PASS
DoS with Revert.	PASS
DoS with block gas limit.	PASS
Methods execution permissions.	PASS
Economy model of the contract.	PASS
The impact of the exchange rate on the logic.	PASS
Private user data leaks.	PASS
Malicious Event log.	PASS
Scoping and Declarations.	PASS
Uninitialized storage pointers.	PASS

Arithmetic accuracy.	PASS
Design Logic.	PASS
Cross-function race conditions.	PASS
Safe Open Zeppelin contracts implementation and usage.	PASS
Fallback function security.	PASS

Audit Findings

Severity	High
Contract	DAOMarketplace.sol
Description	cancelDirectListing does not transfer the NFT back to the original owner.

Code Snippet	function cancelDirectListing(
	uint256 _listingId
	) external onlyListingCreator(_listingId)
	{
	Listing memory targetListing =
	listings[_listingId];
	
	require(targetListing.listingType ==
	ListingType.Direct, "!DIRECT");
	
	delete listings[_listingId];
	delete
	exsist[targetListing.assetContract]
	[targetListing.tokenId];
	
	emit ListingRemoved(_listingId,
	targetListing.tokenHash ,
	targetListing.tokenOwner);
	}

Recommendation	cancelDirectListing deletes the listing without transferring the asset back to the owner like the cancelAuction function does. Add the transferListing function to the end of cancelDirectListing.
Status	

Audit Findings

Severity	**Medium**
Contract	DAOMarketplace.sool
Description	AssetContract does not implement IERC2981 royaltyInfo.
Code Snippet	try IERC2981Upgradeable(_listing.assetContract).royaltyInfo(_listing.tokenId, _totalPayoutAmount)
Recommendation	Underlying asset contract does not implement the function royaltyInfo and therefore can't pay royalties. It's been pushed into a try block, so I am assuming the devs already know this. The only way to really do anything about this is to redeploy the NFT contract if absolutely necessary.
Status	

Audit Findings

Severity	Low(multiple)
Contract	DAOMarketplace.sol
Description	Strict equivalence
Code Snippet	N/A
Recommendation	In general, strict equivalencies should be avoided. Example: function captureListingFee(address _currencyToUse) internal { require(msg.value == listingFee, "!FEE"); msg.value can instead be >= listingFee, which will help avoid any failures due to slight rounding errors.
Status	

Functional Test Status

Function Name	Type/Return Type	Score
_msgData	internal	PASS
_msgSender	internal	PASS
initialize	internal	PASS
supportsInterface	public	PASS
__ERC2771Context_init	internal	PASS
receive	external	PASS
captureListingFee	internal	PASS
onERC721Received	external	PASS
createSell	write/external	PASS
createAuction	write/external	PASS
updateListing	write/external	PASS
_updateListing	internal	PASS
cancelDirectListing	external	FAIL
executeSale	internal	PASS
handleOffer	internal	PASS
handleBid	internal	PASS
isNewWinningBid	internal	PASS
closeAuction	write/external	PASS

Function Name	Type/Return Type	Score
deposit	external	PASS
withdraw	external	PASS
transferCurrency	internal	PASS
transferCurrencyWithWrapper	internal	PASS
safeTransferNativeToken	internal	PASS
safeTransferNativeTokenWithWrapper	internal	PASS
isTrustedForwarder	read/public	PASS
getPlatformFeeInfo	read/external	PASS
setPlatformFeeInfo	write/external	PASS
contractType	external	PASS
contractVersion	external	PASS
contractURI	external	PASS
updateListing	write/external	PASS
cancelDirectListing	write/external	PASS
buy	write/external	PASS
offer	write/external	PASS
accept offer	write/external	PASS
closeAuction	write/external	PASS

Function Name	Type/Return Type	Score
name	read/public	PASS
symbol	read/public	PASS
totalSupply	read/public	PASS
transferFrom	write/public	PASS
_baseURI	internal	PASS
airdrop	write/public	PASS
mint	write/external	PASS
setBaseExtension	write/public	PASS
setBaseURI	write/public	PASS
setHiddenMetadataURI	write/public	PASS
setMax	write/public	PASS
setMaxSupply	write/public	PASS
setMerkleRoot	write/public	PASS
setPrice	write/public	PASS
setReveal	write/public	PASS
setSale	write/public	PASS
tokenURI	read/public	PASS
whitelistMint	read/public	PASS

_cancelAuction	internal	PASS
_closeAuctionForAuctionCreator	internal	PASS
_closeAuctionForBidder	internal	PASS
checkAlreadyListing	internal	PASS
transferListingToken	internal	PASS
payout	internal	FAIL
validateERC20BalanceAndAllowance	internal	PASS
validateOwnershipAndApproval	internal	PASS
validateDirectlistingSale	internal	PASS
getSafeQuantity	internal	PASS
getTokenType	internal	PASS
getPlatformFeeInfo	read/external	PASS
setPlatformFeeInfo	write/external	PASS
allowContract	write/external	PASS
denieContract	write/external	PASS
setListingFee	write/external	PASS
setAuctionBuffers	write/external	PASS
setContractURI	write/external	PASS

Function Name	Type/Return Type	Score
_disableInitializers	internal	PASS
_getInitializedVersion	internal	PASS
_isInitializing	internal	PASS
__Multicall_init	internal	PASS
__Multicall_init_unchained	internal	PASS
multicall	external	PASS
_functionDelegateCall	private	PASS
__ReentrancyGuard_init	internal	PASS
__ReentrancyGuard_init_unchained	internal	PASS
_nonReentrantBefore	private	PASS
_nonReentrantAfter	private	PASS
grantRole	write/external	PASS
revokeRole	write/external	PASS
renounceRole	write/external	PASS
getRoleMember	read/external	PASS
getRoleMemberCount	read/external	PASS
hasRole	read/public	PASS
_checkRole	internal	PASS
getRoleAdmin	read/public	PASS

Function Name	Type/Return Type	Score
_setupRole	internal	PASS
_setRoleAdmin	internal	PASS
royaltyInfo	read/external	PASS
onERC1155Received	external	PASS
onERC1155BatchReceived	external	PASS
allowance	read/external	PASS
isContract	internal	PASS
sendValue	internal	PASS
functionCall	internal	PASS
functionCallWithValue	internal	PASS
functionStaticCall	internal	PASS
functionDelegateCall	internal	PASS
verifyCallResult	internal	PASS
safeTransfer	internal	PASS
safeTransferFrom	internal	PASS
safeApprove	internal	PASS
safeIncreaseAllowance	internal	PASS
safeDecreaseAllowance	internal	PASS
_callOptionalReturn	private	PASS

Automated Review

Ascendant

Conclusion

The smart contracts reviewed in this audit contain no critical severity issues and all Medium to Low issues have either been corrected or acknowledged.

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.

Ascendant

Ascendant

@ascendantproj

www.ascendant.finance

Ascendant