Smart Contract

Security Assessment

For LifestyleDAO

01 Feb 2023

Ascendant

Ascendant

@ascendantproj

www.ascendant.finance

Ascendant

Table of Contents

DISCLAIMER

Executive Summary

Severity	Found
🔴 High	1
🟠 Medium	1
🟡 Low	5
🟣 Informational	49
Total	56

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 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	LifestyleDAO
Platform	Ethereum
Language	Solidity

Contracts Assessed

Name	Location
LifestyleDAO.sol	Goerli: 0x12A1284f6F1a5bDcC87198ca1eCF92ABe3e3795C
ERC721A.sol	In LifestyleDAO contract
Ownable.sol	In LifestyleDAO contract
Context.sol	In LifestyleDAO contract
Address.sol	In LifestyleDAO contract
IERC721Receiver.sol	In LifestyleDAO contract

Name	Location
IERC165.sol	In LifestyleDAO contract
ERC165.sol	In LifestyleDAO contract
IERC721.sol	In LifestyleDAO contract
IERC721Metadata.sol	In LifestyleDAO contract
Strings.sol	In LifestyleDAO contract
Merkleproof.sol	In LifestyleDAO contract

Findings Summary

Severity	Found
🔴 High	1
🟠 Medium	1
🟡 Low	5
🟣 Informational	49
Total	

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

Ascendant

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	LifestyleDAO.sol
Description	Exposed baseURI
Code Snippet	string public baseURI;
Recommendation	Marking the baseURI public allows bad actors to obtain the location of the metadata and download all NFTs, even the ones that have not yet been minted. To prevent this, the baseURI variable should be set to private, so when owner sets baseURI after deployment the baseURI is visible to no one.
Status	

Audit Findings

Severity	**Medium**
Contract	LifestyleDAO.sol
Description	Max supply can be changed, but collection size is immutable.

Code Snippet	function setMaxSupply(uint256 _amount) public onlyOwner {
	 maxSupply = _amount;
	 }

Recommendation	Once a collection is committed to IPFS with a certain number of tokenIDs, this collection cannot be changed under the same address. Furthermore, if a new baseTokenURI is set anytime during or after the initial mint, the new IPFS hash will not change what is already in a holder's wallet. If the total collection size is currently not known, it is recommended to only use this function to set the max supply once the total collection size becomes known and then never called again.
Status	

Audit Findings

Severity	Low
Contract	LifestyleDAO.sol
Description	revealed variable is hardcoded to true
Code Snippet	bool public revealed = true;
Recommendation	Reveal functions work by returning placeholder metadata when the tokenURI function is called. However, once this variable is set to true and the tokenURI function reads the true baseURI, this is committed to the blockchain and cannot be reversed. Because of this, the reveal funciton is a one-shot deal. Once NFTs are revealed, they cannot be "unrevealed" in someone's wallet. Therefore, if you have the revealed variable set to true at deployment, it is absolutely imperative that you remember to call setReveal and set it to false BEFORE any NFTs are minted.
Status	

Audit Findings

Severity	**Informational**
Contract	LifestyleDAO.sol
Description	misleading error messages

Code Snippet	function whitelistMint(uint256 quantity, bytes32[] calldata _merkleProof)
	 public
	 payable
	 {
	 ...
	 require(preSale, "The contract is paused!");
	...}
	
	function mint(uint256 quantity) external payable {
	 ...
	 require(publicSale, "The contract is paused!");
	...}

Recommendation	publicSale and preSale booleans have the same require statement error message, meaning if a mint fails, it will difficult to troubleshoot. They are also misleading. If preSale is true and publicSale is false, it doesn't mean the contract is actually paused, as whitelisted addresses can still mint and vice versa. Either create an actual variable to pause and unpause the function or revise the error messages to prevent confusion.
Status	

Audit Findings

Severity	**Informational**
Contract	LifestyleDAO.sol
Description	Public functions that are used externally and not by the contract itself should be marked external
Code Snippet	N/A
Recommendation	Public functions generally consume more gas than external functions. Any functions that are not used internally should be marked external.
Status	

Functional Test Status

Function Name	Type/Return Type	Score
_msgData	internal	PASS
_msgSender	internal	PASS
_baseURI	internal	PASS
_checkContractOnERC721Received	private	PASS
_exists	internal	PASS
_getAux	internal	PASS
_numberMinted	internal	PASS
_numberBurned	internal	PASS
_ownershipOf	internal	PASS
_safeMint	internal	PASS
_setAux	intenral	PASS
_startTokenId	internal	PASS
_totalMinted	internal	PASS
ownerOf	read/public	PASS
balanceOf	read/public	PASS
getApproved	write/public	PASS
isApprovedForAll	read/public	PASS
safeTransferFrom	write/public	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

withdraw	write/public	PASS
owner	read/public	PASS
renounceOwnership	write/public	PASS
transferOwnership	write/public	PASS
increaseAllowance	write/public	PASS
decreaseAllowance	write/public	PASS
_transfer	internal	PASS
_mint	internal	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