

Smart Contract Security Assessment

For Resilient Her
23 Sept 2022



Ascendant

Ascendant

@ascendantproj
www.ascendant.finance



Ascendant

Table of Contents

3 Disclaimer

4 Executive Summary

5 Overview

6 Findings Summary & Legend

8 Manual Review

- Issue Checking Status
- Audit Findings
- Functional Test Status
- Omitted Results

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 advocacy 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	8
● Informational	69
Total	79

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 Resilient Her 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	ResilientHer
Platform	Ethereum
Language	Solidity

Contracts Assessed





Name	Location
ResilientHer.sol	Not Published
Counters.sol	In ResilientHer contract
Merkleproof.sol	In ResilientHer contract
IERC20permit.sol	In ResilientHer contract
IERC20.sol	In ResilientHer contract
ReentrancyGuard.sol	In ResilientHer contract

Name	Location
Strings.sol	In ResilientHer contract
Context.sol	In ResilientHer contract
Ownable.sol	In ResilientHer contract
Address.sol	In ResilientHer contract
SafeERC20.sol	In ResilientHer contract
PaymentSplitter.sol	In ResilientHer contract
IERC721Receiver.sol	In ResilientHer contract
IERC165.sol	In ResilientHer contract
ERC165.sol	In ResilientHer contract
IERC721.sol	In ResilientHer contract
IERC721Metadata.sol	In ResilientHer contract
ERC721.sol	In ResilientHer contract
OwnableDelegateProxy.sol	In ResilientHer contract
ProxyRegistry.sol	In ResilientHer contract

Findings Summary

Severity	Found
 High	1
 Medium	1
 Low	8
 Informational	69
Total	79

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	ResilientHer.sol
Description	Exposed baseURI
Code Snippet	<pre>string public baseURI = [REDACTED]</pre>
Recommendation	<p>Writing the baseURI into the smart contract 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, and the constructor should accepted a string argument that will allow the owner to set the baseURI on deployment so the baseURI is visible to no one.</p>
Status	AMENDED

Audit Findings

Severity	Medium
Contract	ResilientHer.sol
Description	Checks-Effects-Interactions
Code Snippet	<pre>function presaleMint(... _presaleClaimed[msg.sender] += _amount; for (uint256 i = 0; i < _amount; i++) { mintInternal(); } }</pre>
Recommendation	<p>The mint functions currently all fail the checks-effects-interactions pattern, which requires that interactions (such as <code>_safeMint</code>) should come after state changes (updating the mapping), which exposes the function to Reentrancy attacks.</p> <p><code>_presaleClaimed[msg.sender] += _amount</code> should either precede <code>_safeMint</code> or a <code>ReentrancyGuard</code> should be added to these functions.</p>
Status	AMENDED: nonReentrant modifier added to internal mint function

Audit Findings

Severity	Low
Contract	ResilientHer.sol
Description	PaymentSplitter is unnecessary/not utilized.
Code Snippet	<pre>uint256[] private _teamShares = [100];</pre>
Recommendation	Currently, teamShares is set to 100% and there is only one address initialized with the contract. PaymentSplitter does not need to be used.
Status	AMENDED: PaymentSplitter removed and withdraw function added

Audit Findings

Severity	Low
Contract	ResilientHer.sol
Description	No setPrice functions
Code Snippet	None
Recommendation	<p>There are no setPrice functions, which means that after deployment, there will be no way to change prices if necessary. If plans change, the contract will need to be redeployed.</p> <p>Never hardcode values or leave out set functions that may be needed in the future.</p>
Status	AMENDED

Functional Test Status

Function Name	Type/Return Type	Score
_addPayee	private	PASS
_pendingPayment	private	PASS
payee	read/public	PASS
release	write/public	PASS
released	read/public	PASS
setRoyalties	write/external	PASS
totalReleased	read/public	PASS
totalShares	read/public	PASS
updatePayee	write/public	PASS
_setRoyalties	internal	PASS
owner	read/public	PASS
renounceOwnership	write/public	PASS
transferOwnership	write/public	PASS
transferFrom	write/public	PASS
increaseAllowance	write/public	PASS
decreaseAllowance	write/public	PASS
_transfer	internal	PASS
_mint	internal	PASS

Function Name	Type/Return Type	Score
Counters		
current	internal	PASS
decrement	internal	PASS
increment	internal	PASS
reset	internal	PASS
MerkleProof		
verify	internal	PASS
processProof	internal	PASS
IERC20Permit		
permit	write/external	PASS
IERC20		
allowance	read/external	PASS
approve	write/external	PASS
balanceOf	read/external	PASS
totalSupply	read/external	PASS
transfer	write/external	PASS
transferFrom	write/external	PASS
Context		

Function Name	Type/Return Type	Score
_msgData	internal	PASS
_msgSender	internal	PASS
Ownable		
_checkOwner	internal	PASS
transferOwnership	write/public	PASS
owner	read/public	PASS
renounceOwnership	write/public	PASS
Address		
functionCall	internal	PASS
functionCallWithValue	internal	PASS
functionDelegateCall	internal	PASS
functionStaticCall	internal	PASS
isContract	internal	PASS
sendValue	internal	PASS
verifyCallResult	internal	PASS
SafeERC20		
safeApprove	internal	PASS
safeDecreaseAllowance	internal	PASS

safeIncreaseAllowance	internal	PASS
safePermit	internal	PASS
safeTransfer	internal	PASS
safeTransferFrom	internal	PASS
PaymentSplitter		
payee	read/public	PASS
receive	external	PASS
releasable	read/public	PASS
release	write/public	PASS
released	read/public	PASS
shares	read/public	PASS
totalReleased	read/public	PASS
totalShares	read/public	PASS
IERC721Receiver		
onERC721Received	external	PASS
IERC721		
approve	write/external	PASS
balanceOf	read/external	PASS
getApproved	read/external	PASS
isApprovedForAll	read/external	PASS
ownerOf	read/external	PASS
safeTransferFrom	write/external	PASS

setApprovalForAll	write/external	PASS
transferFrom	write/external	PASS
ERC721		
symbol	read/public	PASS
tokenURI	read/public	PASS
name	read/public	PASS
ResilientHer		
isApprovedForAll	read/public	PASS
publicSaleMint	write/external	PASS
reveal	write/public	PASS
setBaseExtension	write/public	PASS
setBaseURI	write/public	PASS
setMerkleRoot	write/public	PASS
setNotRevealedURI	write/public	PASS
togglePause	write/public	PASS
togglePresale	write/public	PASS
tokenURI	read/public	PASS
totalSupply	read/public	PASS

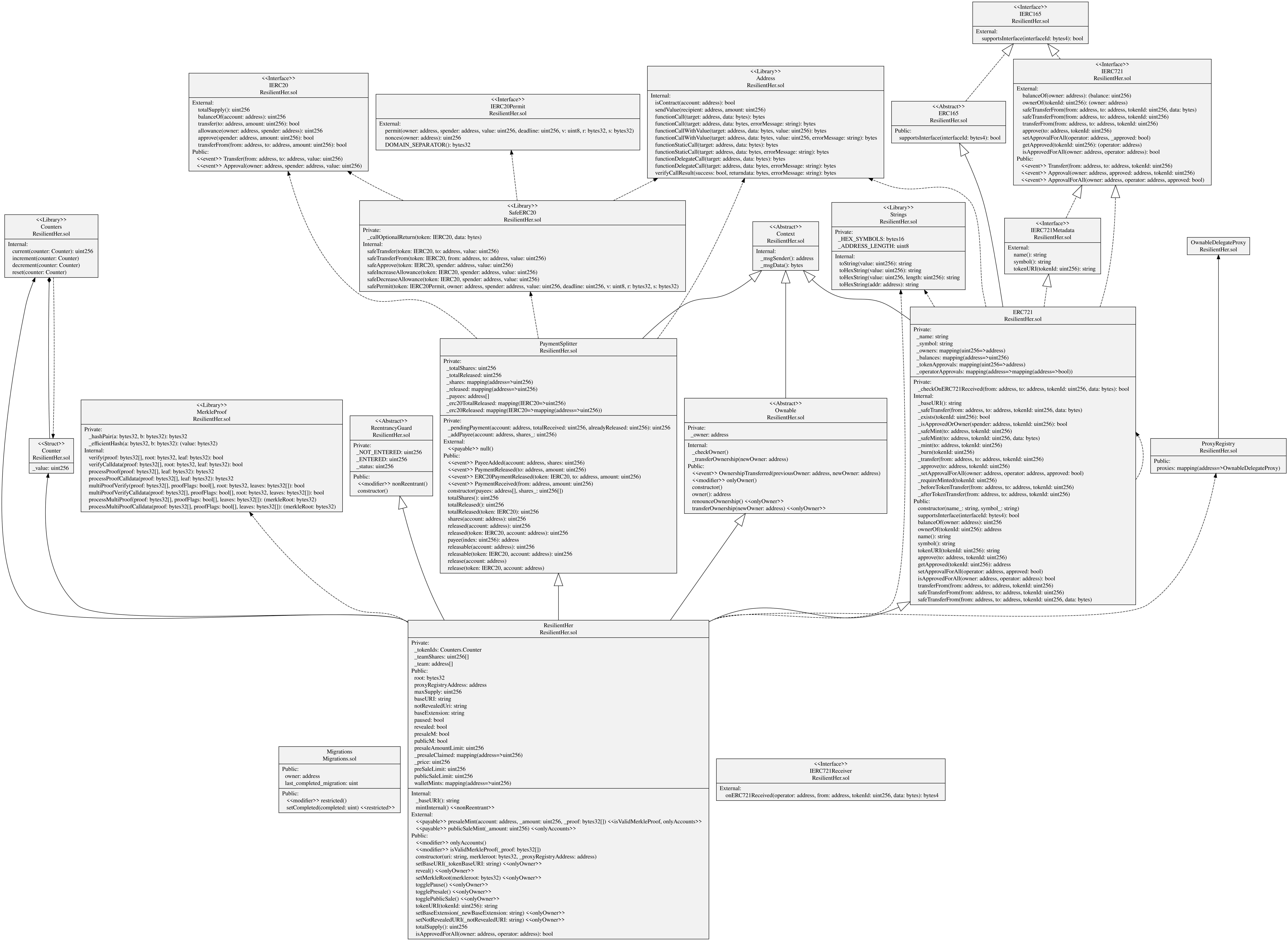
Omitted Results

Note: Any issues that have been omitted from this report have been deemed by the reviewing team as irrelevant, inapplicable, and/or negligible to the proper functioning of this contract. Thus, any omitted issues can be safely ignored.

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