



February 13, 2025


# Express Audit Report for SinWar [SW]

DISCLAIMER: This is an automatically generated audit performed with De.Fi Scanner tool. De.Fi smart contract auditing tool is intended to assist in identifying potential vulnerabilities or malicious functions in smart contracts. While this is done to our best effort and knowledge, please notice that no tool can guarantee complete accuracy or comprehensiveness in detecting all possible vulnerabilities.



## Project Summary


Project Name	SinWarCoin
Address	<a href="#">0x24cc192950aaac600dc886ad0005ffe98a5971ff</a>
Network	56

Issue ID	103
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<b>pragma solidity ^0.8.0;</b>
Location	<p>Different versions of Solidity is used:</p> <ul style="list-style-type: none"> <li>- Version used: ['^0.8.0', '^0.8.1', '^0.8.28']</li> <li>- ^0.8.0 (Ownable.sol#4)</li> <li>- ^0.8.0 (Ownable2Step.sol#4)</li> <li>- ^0.8.0 (ERC20.sol#4)</li> <li>- ^0.8.0 (IERC20.sol#4)</li> <li>- ^0.8.0 (IERC20Metadata.sol#4)</li> <li>- ^0.8.0 (IERC20Permit.sol#4)</li> <li>- ^0.8.0 (SafeERC20.sol#4)</li> <li>- ^0.8.1 (Address.sol#4)</li> <li>- ^0.8.0 (Context.sol#4)</li> <li>- ^0.8.1 (Address.sol#4)</li> <li>- ^0.8.0 (Context.sol#4)</li> <li>- ^0.8.0 (ERC20.sol#4)</li> <li>- ^0.8.0 (IERC20.sol#4)</li> <li>- ^0.8.0 (IERC20Metadata.sol#4)</li> <li>- ^0.8.0 (IERC20Permit.sol#4)</li> <li>- ^0.8.0 (Ownable.sol#4)</li> <li>- ^0.8.0 (Ownable2Step.sol#4)</li> <li>- ^0.8.0 (SafeERC20.sol#4)</li> <li>- ^0.8.28 (SinWarCoin.sol#2)</li> <li>- ^0.8.0 (IERC20Metadata.sol#4)</li> <li>- ^0.8.0 (IERC20Permit.sol#4)</li> <li>- ^0.8.1 (Address.sol#4)</li> <li>- ^0.8.0 (Context.sol#4)</li> </ul>

### SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This will be an important consideration for a future version update of the contract if bugs become evident anytime leading up to and during the main launch event.


**IMPORTANT NOTE:** The smart contract was compiled using the latest version as of the 13/02/2025; v0.8.28+commit.7893614a.

Issue ID	177
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<code>pragma solidity ^0.8.0;</code>
Location	Pragma version^0.8.0 (Ownable.sol#4) allows old versions

### SW - Developer Comment:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This will be an important consideration for a future version update of the contract if bugs become evident anytime leading up to and during the main launch event.


IMPORTANT NOTE: The smart contract was compiled using the latest version as of the 13/02/2025; v0.8.28+commit.7893614a.

Issue ID	177
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<b>pragma solidity ^0.8.1;</b>
Location	Pragma version^0.8.1 (Address.sol#4) allows old versions

### SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This will be an important consideration for a future version update of the contract if bugs become evident anytime leading up to and during the main launch event.

IMPORTANT NOTE: The smart contract was compiled using the latest version as of the 13/02/2025; v0.8.28+commit.7893614a.

Issue ID	177
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<b>pragma solidity ^0.8.28;</b>
Location	Pragma version^0.8.28 (SinWarCoin.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

## SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This will be an important consideration for a future version update of the contract if bugs become evident anytime leading up to and during the main launch event.

IMPORTANT NOTE: The smart contract was compiled using the latest version as of the 13/02/2025; v0.8.28+commit.7893614a.




Issue ID	177
Severity	🟡 Informational
Status	High
Description Code	
Location	solc-0.8.28 is not recommended for deployment

### SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, this is the version recommended as it's the latest version which includes any bug fixes identified in previous versions. This will be an important consideration for a future version update of the contract if bugs become evident anytime leading up to and during the main launch event.


IMPORTANT NOTE: The smart contract was compiled using the latest version as of the 13/02/2025; v0.8.28+commit.7893614a.

Issue ID	173
Severity	 Informational
Status	High
Description Code	<pre>function _callOptionalReturnBool(IERC20 token, bytes memory data) private returns (bool) { // We need to perform a low level call here, to bypass Solidity's return data size checking mechanism, since // we're implementing it ourselves. We cannot use {Address-functionCall} here since this should return false // and not revert is the subcall reverts. (bool success, bytes memory returndata) = address(token).call(data); return success &amp;&amp; (returndata.length == 0    abi.decode(returndata, (bool))) &amp;&amp; Address.isContract(address(token)); }</pre>
Location	<p>Low level call in SafeERC20._callOptionalReturnBool(IERC20,bytes) (SafeERC20.sol#134-142): - (success,returndata) = address(token).call(data) (SafeERC20.sol#139)</p>

## SW - Developer Comments:


- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to SafeERC20.sol file import.



Issue ID	173
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<pre><b>function</b> <b>sendValue</b>(<b>address payable</b> recipient, <b>uint256</b> amount) <b>internal</b> {   <b>require</b>(<b>address</b>(<b>this</b>).<b>balance</b> &gt;= amount, "Address: insufficient balance");   (<b>bool</b> success, ) = recipient.<b>call</b>{<b>value</b>: amount}("");   <b>require</b>(success, "Address: unable to send value, recipient may have reverted"); }</pre>
Location	<p>Low level call in Address.sendValue(address,uint256) (Address.sol#64-69):</p> <ul style="list-style-type: none"><li>- (success) = recipient.call{value: amount}() (Address.sol#67)</li></ul>


## SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to file Address.sol which the SafeERC20.sol file uses as an import.

Issue ID	173
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<pre>function <b>functionCallWithValue</b>(   address target,   bytes memory data,   uint256 value,   string memory errorMessage ) <b>internal returns</b> (bytes memory) {   require(address(this).balance &gt;= value, "Address:   insufficient balance for call");   (bool success, bytes memory returndata) =   target.call{value: value}(data);   return verifyCallResultFromTarget(target, success,   returndata, errorMessage); }</pre>
Location	<p>Low level call in Address.functionCallWithValue(address,bytes,uint256 ,string) (Address.sol#128-137): - (success,returndata) = target.call{value: value}(data) (Address.sol#135)</p>


## SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to file Address.sol which the SafeERC20.sol file uses as an import.

Issue ID	173
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<pre><b>function</b> <b>functionStaticCall</b>(   <b>address</b> target,   <b>bytes memory</b> data,   <b>string memory</b> errorMessage ) <b>internal view returns</b> (<b>bytes memory</b>) {   (<b>bool</b> success, <b>bytes memory</b> returndata) =   target.<b>staticcall</b>(data);   <b>return</b> verifyCallResultFromTarget(target, success,   returndata, errorMessage); }</pre>
Location	<p>Low level call in Address.functionStaticCall(address,bytes,string) (Address.sol#155-162): - (success,returndata) = target.staticcall(data) (Address.sol#160)</p>


## SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to file Address.sol which the SafeERC20.sol file uses as an import.

Issue ID	173
Severity	 <b>Informational</b>
Status	<b>High</b>
Description Code	<pre>function <b>functionDelegateCall</b>(   address target,   bytes memory data,   string memory errorMessage ) <b>internal returns</b> (bytes memory) {   (bool success, bytes memory returndata) =   target.delegatecall(data);   return verifyCallResultFromTarget(target, success,   returndata, errorMessage); }</pre>
Location	<p>Low level call in Address.functionDelegateCall(address,bytes,string) (Address.sol#180-187): - (success,returndata) = target.delegatecall(data) (Address.sol#185)</p>

## SW - Developer Comments:

- This is highlighted as informational however it is an important consideration, due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to file Address.sol which the SafeERC20.sol file uses as an import.

Issue ID	168
Severity	 Low
Status	Medium
Description Code	<b>function transferOwnership(address newOwner) public virtual override onlyOwner {</b>
Location	Ownable2Step.transferOwnership(address).newOwner (Ownable2Step.sol#35) lacks a zero-check on : - _pendingOwner = newOwner (Ownable2Step.sol#36)


## SW - Developer Comments:

- Due to imports of open-sourced libraries this creates numerous dependencies. This is a functional import from openzeppelin library, more specifically related to file Ownable2Step.sol which the main Sinwarcoin.sol uses as an import. The contract has been deployed successfully with the owner being the development teams wallet address, however if a bug is detected in the future this will be an important consideration.

Issue ID	189
Severity	🔴 Critical
Status	High
Description Code	<pre><b>function transfer</b>(address to, uint256 amount) <b>public virtual override returns</b> (bool) {   address owner = _msgSender();   _transfer(owner, to, amount);   <b>return</b> true; }</pre>
Location	<p>Pausable function: ERC20.transfer(address,uint256) (ERC20.sol#113-117)</p> <ul style="list-style-type: none"><li>- in internal call: _transfer</li><li>- In expression: require(bool,string)(tradingEnabled    from == owner(),Trading disabled)</li></ul>


### SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this gives the development team the power to disable or enable trading in case of technical emergencies to protect against unforeseen circumstances in the cryptosphere. This operates as a safety switch.

Issue ID	189
Severity	 <b>Critical</b>
Status	<b>High</b>
Description Code	<pre> <b>function transferFrom</b>(address from, address to, <b>uint256</b> amount) <b>public virtual override returns</b> <b>(bool)</b> {   <b>address</b> spender = _msgSender();   _spendAllowance(<b>from</b>, spender, amount);   _transfer(<b>from</b>, to, amount);   <b>return true</b>; } </pre>
Location	<p>Pausable function:  ERC20.transferFrom(address,address,uint256)  (ERC20.sol#158-163)</p> <ul style="list-style-type: none"> <li>- in internal call:_transfer</li> <li>- In expression: require(bool,string)(tradingEnabled    from == owner(),Trading disabled)</li> </ul>

## SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this gives the development team the power to disable or enable trading in case of technical emergencies to protect against unforeseen circumstances in the cryptosphere. This operates as a safety switch.

Issue ID	209
Severity	 Critical
Status	High
Description Code	<pre>function transfer(address to, uint256 amount) public virtual override returns (bool) {     address owner = _msgSender();     _transfer(owner, to, amount);     return true; }</pre>
Location	<p>Transfer Fee: ERC20.transfer(address,uint256) (ERC20.sol#113-117)</p> <ul style="list-style-type: none"><li>- in nested function: _transfer</li><li>- in expression: (amount * charityFee) / 10000</li></ul>

## SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this is one of the core features of the project where a small fee (charity fee) on every transaction is pooled back into the the development's team wallet. The fee can be variable between a minimum of 0.5% to a maximum of 2.5%. All proceeds accumulated in the developer wallet will go to support the reconstruction of Gaza.



Issue ID	7
Severity	🎯 Data
Status	High
Description Code	
Location	Transfer fee variables

### SW - Developer Comments:


- This is a function implemented in the main Sinwarcoin.sol file, this is one of the core features of the project where a small fee (charity fee) on every transaction is pooled back into the the development's team wallet. The fee can be variable between a minimum of 0.5% to a maximum of 2.5%. All proceeds accumulated in the developer wallet will go to support the reconstruction of Gaza.



Issue ID	8
Severity	🎯 Data
Status	High
Description Code	
Location	Transfer fee limits


### SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this is one of the core features of the project where a small fee (charity fee) on every transaction is pooled back into the the development's team wallet. The fee can be variable between a minimum of 0.5% to a maximum of 2.5%. All proceeds accumulated in the developer wallet will go to support the reconstruction of Gaza.

Issue ID	211
Severity	 Critical
Status	High
Description Code	<pre>function transfer(address to, uint256 amount) public virtual override returns (bool) { address owner = _msgSender(); _transfer(owner, to, amount); return true; }</pre>
Location	<p>Transfer amount limits in:</p> <p>ERC20.transfer(address,uint256) (ERC20.sol#113-117)</p> <ul style="list-style-type: none"><li>- In expression: fromBalance &gt;= amount</li><li>- In expression: netAmount &lt;= maxTxAmount</li></ul>

## SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this feature is incorporated into the contract to prevent any whale dumping that may occur which would result in significant price fluctuations. These limits can vary depending on the project phase.

Issue ID	211
Severity	 <b>Critical</b>
Status	<b>High</b>
Description Code	<pre> <b>function transferFrom</b>(address from, address to, <b>uint256</b> amount) <b>public virtual override returns</b> <b>(bool)</b> {   <b>address</b> spender = _msgSender();   _spendAllowance(<b>from</b>, spender, amount);   _transfer(<b>from</b>, to, amount);   <b>return true</b>; } </pre>
Location	<p>Transfer amount limits in:  ERC20.transferFrom(address,address,uint256)  (ERC20.sol#158-163)</p> <ul style="list-style-type: none"> <li>- In expression: fromBalance &gt;= amount</li> <li>- In expression: netAmount &lt;= maxTxAmount</li> <li>- In expression: currentAllowance &gt;= amount</li> </ul>

### SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this feature is incorporated into the contract to prevent any whale dumping that may occur which would result in significant price fluctuations. These limits can vary depending on the project phase.



Issue ID	6
Severity	🎯 Data
Status	High
Description Code	
Location	Transfer limits


### SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this feature is incorporated into the contract to prevent any whale dumping that may occur which would result in significant price fluctuations. These limits can vary depending on the project phase.

Issue ID	237
Severity	🟡 Low
Status	High
Description Code	<pre>function transfer(address to, uint256 amount) public virtual override returns (bool) {     address owner = _msgSender();     _transfer(owner, to, amount);     return true; }</pre>
Location	<p>whitelisted function: ERC20.transfer(address,uint256) (ERC20.sol#113-117)</p> <p>- in internal call: SinWarCoin._transfer(address,address,uint256) (SinWarCoin.sol#56-86)</p> <p>- in expression ! isExcludedFromFees[from] &amp;&amp; ! isExcludedFromFees[to]</p>

## SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this feature is incorporated into the contract to allow selected contract addresses exemptions from any fee's or limits. The principle behind this is the case example of the pre-sale contract address, no charity fee will be applied when interacting with the pre-sale contract address. In addition, as this is a long term vision of the project, when partnering with humanitarian and charity organisations, contract wallets belonging to such organisations would be included in the fee exemptions as it makes no sense to charge a charity fee on a charity organisation.

Issue ID	237
Severity	 <b>Low</b>
Status	<b>High</b>
Description Code	<pre> function <b>transferFrom</b>(address from, address to, uint256 amount) <b>public virtual override returns</b> (bool) {     address spender = _msgSender();     _spendAllowance(<b>from</b>, spender, amount);     _transfer(<b>from</b>, to, amount);     <b>return</b> true; } </pre>
Location	<p>whitelisted function:</p> <p>ERC20.transferFrom(address,address,uint256) (ERC20.sol#158-163)</p> <p>- in internal call:</p> <p>SinWarCoin._transfer(address,address,uint256) (SinWarCoin.sol#56-86)</p> <p>- in expression ! isExcludedFromFees[from] &amp;&amp; ! isExcludedFromFees[to]</p>

## SW - Developer Comments:

- This is a function implemented in the main Sinwarcoin.sol file, this feature is incorporated into the contract to allow selected contract addresses exemptions from any fee's or limits. The principle behind this is the case example of the pre-sale contract address, no charity fee will be applied when interacting with the pre-sale contract address. In addition, as this is a long term vision of the project, when partnering with humanitarian and charity organisations, contract wallets belonging to such organisations would be included in the fee exemptions as it makes no sense to charge a charity fee on a charity organisation.