Audit Report

Werewolves of Wall Street

February 2022

Type	ERC-721
Network	ETH
Address	0x3Fd24DF913dA5bD7FEC34208DE2A95f55ec55E44
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	Wolf
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://etherscan.io/token/0x3Fd24DF913dA5bD7FEC34208DE2A95f55ec55E44
Symbol	WOLF
Total Supply	10,000
Source	contract.sol
Domain	werewolvesofwallstreet.io

Audit Updates

Initial Audit	21st February 2022
Corrected	

Contract Analysis

- Users have the ability to mint many nfts in one transaction.
- The mint cost of the NFT varies between the pre-sale and post-sale state.
- These two NFT mint costs can be changed by the contract owner.
- The maximum number of NFTs that can be created is 10,000.
- The amount that is accumulated to the contract from the payments, can be withdrawn from the contract owner.
- The contract owner has the ability to set the presale wallet addresses.
- When the presale is enabled, only the presale wallets can mint NFTs.
- The contact owner has the ability to toggle the presale state without limit.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CO	Code Optimization
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L12	Using Variables before Declaration
	L07	Missing Events Arithmetic
	L14	Uninitialized Variables in Local Scope

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1363

Description

The function is aiming to return the token ids that are holded by the specific owner. The ERC721Enumerable class contains a variable called *_ownedTokens* that keeps the token ids per user address. This structure contains exactly the same information that the *walletOfOwner* is calculating.

```
function walletOfOwner(address _owner)
    public
    view
    returns (uint256[] memory)
{
    uint256 ownerTokenCount = balanceOf(_owner);
    uint256[] memory tokenIds = new uint256[](ownerTokenCount);
    for (uint256 i; i < ownerTokenCount; i++) {
        tokenIds[i] = tokenOfOwnerByIndex(_owner, i);
    }
    return tokenIds;
}
```

Recommendation

The walletOfOwner could merely return the information from the `_ownedTokens[_owner]` structure

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L152,160,774,781,788,807,831,845,859,1196 and 11 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
withdraw
removePresaleUser
addMultiplePresaleUser
addPresaleUser
changePreSale
pause
setBaseExtension
setPresaleCost
setCost
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1323

Description

Constant state variables should be declared constant to save gas.

```
maxSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L874,1341,1363,1402,1406,1410,1414,1421,1430,1438 and 2 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_user
_state
_price
_newBaseExtension
_newBaseURI
_newCost
_owner
_mintAmount
_data
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L262,272,291,305,351,361,324,334,237,378 and 5 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
toHexString
_burn
_baseURI
_msgData
verifyCallResult
sendValue
functionStaticCall
functionDelegateCall
functionCallWithValue
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1341,1421

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
inPreSale == true && presaleWallets[msg.sender] == true
inPreSale == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1090,1092

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
reason
retval
```

Recommendation

The variables should be declared before any usage of them.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1402,1406

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
presaleCost = _newCost
cost = _newCost
```

Recommendation

Emit an event for critical parameter changes.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1370

Description

The are variables that are defined in the local scope and are not initialized.

```
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

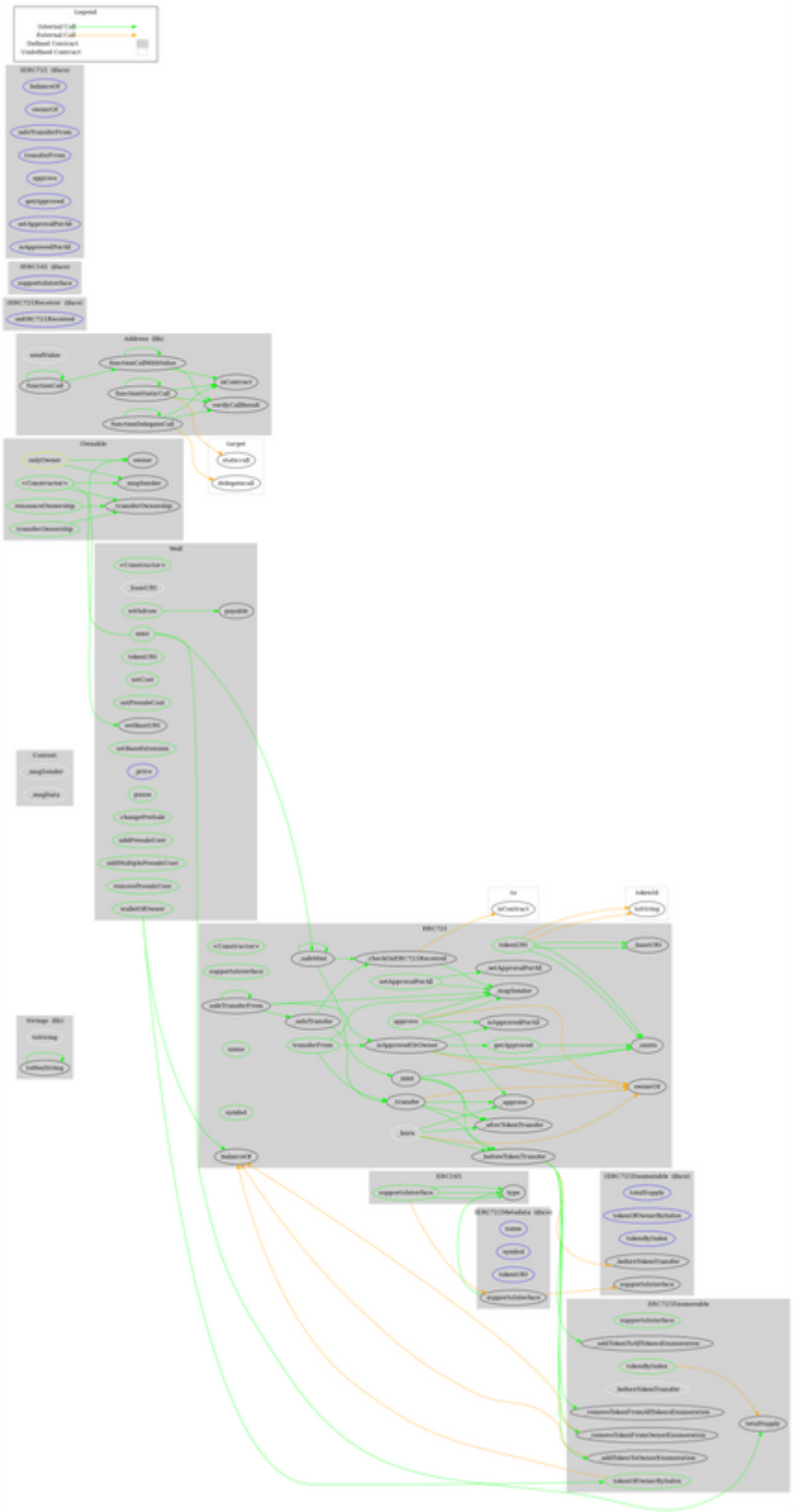
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		

IERC721Receiver	Interface			
	onERC721Received	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata		

	<Constructor>	Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-
	name	Public		-
	symbol	Public		-
	tokenURI	Public		-
	_baseURI	Internal		
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_setApprovalForAll	Internal	✓	
	_checkOnERC721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC721Enumerable	Implementation	ERC721, IERC721Enumerable		
	supportsInterface	Public		-
	tokenOfOwnerByIndex	Public		-
	totalSupply	Public		-
	tokenByIndex	Public		-

	_beforeTokenTransfer	Internal	✓	
	_addTokenToOwnerEnumeration	Private	✓	
	_addTokenToAllTokensEnumeration	Private	✓	
	_removeTokenFromOwnerEnumeration	Private	✓	
	_removeTokenFromAllTokensEnumeration	Private	✓	
Wolf	Implementation	ERC721Enumerable, Ownable		
	<Constructor>	Public	✓	ERC721
	_baseURI	Internal		
	mint	Public	Payable	-
	walletOfOwner	Public		-
	tokenURI	Public		-
	setCost	Public	✓	onlyOwner
	setPresaleCost	Public	✓	onlyOwner
	setBaseURI	Public	✓	onlyOwner
	setBaseExtension	Public	✓	onlyOwner
	_price	External		-
	pause	Public	✓	onlyOwner
	changePreSale	Public	✓	onlyOwner
	addPresaleUser	Public	✓	onlyOwner
	addMultiplePresaleUser	Public	✓	onlyOwner
	removePresaleUser	Public	✓	onlyOwner
	withdraw	Public	Payable	onlyOwner

Contract Flow



Domain Info

Domain Name	werewolvesofwallstreet.io
Registry Domain ID	c86a71e6c6c642b99d63c0d68a0004b1-DONUTS
Creation Date	2021-12-19T15:49:37Z
Updated Date	2021-12-24T15:50:08Z
Registry Expiry Date	2022-12-19T15:49:37Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 2 months before the creation of the audit. It will expire in 10 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Werewolves of Wall Street is an NFT project that implements an additional layer of business logic over the standard IERC-721. The contract supports a presale feature. The presale applicable wallets can mint NFTs at different cost.

The audit mentions the security concerns, contract owner roles, performance improvements and business logic.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Coinscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Coinscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Coinscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Coinscope team disclaims any liability for the resulting losses.

About Coinscope

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co