Smart Contract
Security Audit
V1

The Poker Face Club Smart Contract

27/3/2022

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform**: Ethereum
- **Contract Address**: 0xeFC9d9E65B21fC15F549441EaFa2CB6e57836e02
- **Code:**

https://rinkeby.etherscan.io/address/0xefc9d9e65b21fc15f549441eafa2cb6e57836e02#code

NFT Information

- Name: TPFC
- Total Supply: 4444
- Holders:
- Total transactions:

Contracts address deployed to test net (ETH)

The poker face club Smart contract on ETH test net to test write functions by the auditor.

https://rinkeby.etherscan.io/address/0xefc9d9e65b21fc15f549441eafa2cb6e57836e02

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**. Because the team fix all high and low issues.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 1 high, 0 medium, 1 low, 0 very low-level issues and 1 note in all solidity files of the contract

The files:

ThePokerFace.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
ThePokerFace.sol	**3f7c14fc9964f07e3632266ae17784ed4f127058150aaa8b332c42a8a9f1a249**	0xeFC9d9E65B21fC15F549441EaFa2CB6e57836e02

- Contract: ThePokerFace
- Inherit: ERC721Enumerable, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	**Passed**
symbol	✓	Read / public	**Passed**
SingleWl_isWhiteListed	✓	Read / public	**Passed**
supportsInterface	✓	Read / public	**Passed**
publicSale_price	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
team_counter	✓	Read / public	**Passed**
team_nft	✓	Read / public	**Passed**
getApprovedForAll	✓	Read / public	**Passed**
ownerOf	✓	Read / public	**Passed**
getApproved	✓	Read / public	**Passed**

tokenURI	✓	Read / public	**Passed**
tokenByIndex	✓	Read / public	**Passed**
tokenOfOwnerByIndex	✓	Read / public	**Passed**
tokenOfOwner	✓	Read / public	**Passed**
publicSale_status	✓	Read / public	**Passed**
teamMintSale_status	✓	Read / public	**Passed**
privateSale_supply	✓	Read / public	**Passed**
_baseURI	✓	Read / public	**Passed**
totalSupply	✓	Read / public	**Passed**
_revelNFT	✓	Read / public	**Passed**
getPrice_privateSale	✓	Read / public	**Passed**
contractURI	✓	Read / public	**Passed**
giveaway_counter	✓	Read / public	**Passed**
getPrice_public	✓	Read / public	**Passed**
giveawayMint_status	✓	Read / public	**Passed**
giveawayMint_supply	✓	Read / public	**Passed**
is_WLprivateSale	✓	Read / public	**Passed**
privateSale_counter	✓	Read / public	**Passed**
isteamMember	✓	Read / public	**Passed**
privateSale_price	✓	Read / public	**Passed**
privateSale_status	✓	Read / public	**Passed**
addteamMember	✓	Write / public	**Passed**
approve	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
setBaseURI	✓	Write / public	**Passed**
mint	✓	Write / payable	**Passed**

transferOwnership	✓	Write / public	**Passed**
setApprovalForAll	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**
changeRevelStatus	✓	Write / public	**Passed**
flipPublicSale	✓	Write / public	**Passed**
withdraw	✓	Write / public	**Passed**
flipTeamMint	✓	Write / public	**Passed**
flipWLprivateSale	✓	Write / public	**Passed**
flipgiveaway	✓	Write / public	**Passed**
giveawayMint	✓	Write / public	**Passed**
reserve_nfts	✓	Write / public	**Passed**
privateSale	✓	Write / payable	**Passed**
setPrice_privateSale	✓	Write / public	**Passed**
setPrice_public	✓	Write / public	**Passed**
Single_WLprivateSale	✓	Write / public	**Passed**
SingleWL_giveawaymint	✓	Write / public	**Passed**
teamMint	✓	Write / public	**Passed**
WLteamMint	✓	Write / public	**Passed**
WLgiveawayMint	✓	Write / public	**Passed**
WLprivateSale	✓	Write / public	**Passed**

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed with Notes**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical:

No critical severity vulnerabilities were found.

High:

#Contract code size exceeds 24576 bytes

Description
Contract implementation is too large in size to be deployed on main net. Ethereum with its spurious dragon release limited the size of the contracts deployable on main net to 24576 bytes.
The size of the contract ThePokerFace.sol goes way above this value and
currently is of size 32796bytes.

Remediation
Define and use libraries for pure and view functions e.g. We can create a library which contains all the mathematical operations.

Status: Closed. Fixed in version 2.

Medium:

No Medium severity vulnerabilities were found

Low:

#Multiple pragma statements

Line	Pragma
3	pragma solidity ^0.8.0;
228	pragma solidity ^0.8.0;
293	pragma solidity ^0.8.0;
502	pragma solidity ^0.8.0;
527	pragma solidity ^0.8.0;
550	pragma solidity ^0.8.0;
575	pragma solidity ^0.8.0;
714	pragma solidity ^0.8.0;
739	pragma solidity ^0.8.0;
762	pragma solidity ^0.8.0;
784	pragma solidity ^0.8.0;
851	pragma solidity ^0.8.0;
1225	pragma solidity ^0.8.0;

1413	pragma solidity ^0.8.0;

Description

There are multiple pragma statements in the code. Only the compiler version 0.8.7 will work with the code, but keeping only one pragma statement helps in maintaining readability of the code.

Remediation

Keep a single pragma statement.

Status: Closed. Fixed In version 2

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Naming Conventions

Description

The contract follows a consistent naming convention where we are private variables with leading"_" and public variables without it. But we have missed to comply to the condition for certain variable names "__revelNFT" which is public.

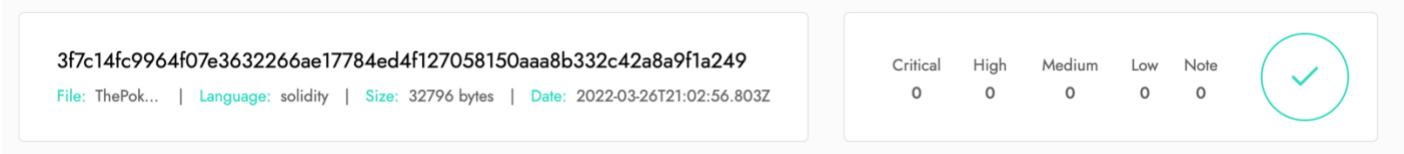
Remediation

Remove "_" from external variable names and add it to private variable names.

Status: Closed. Fixed In version 2

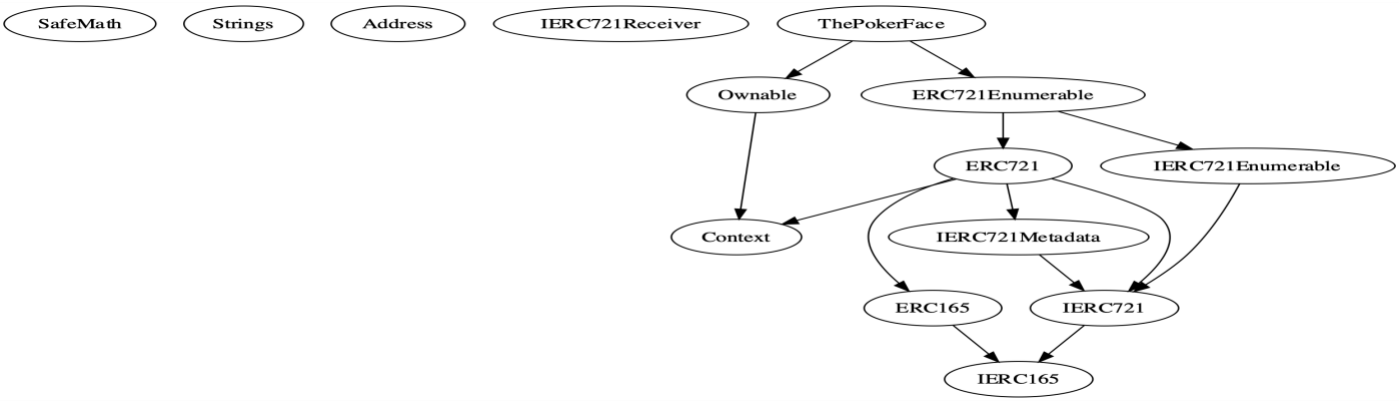
Automatic Testing

1- Check for security

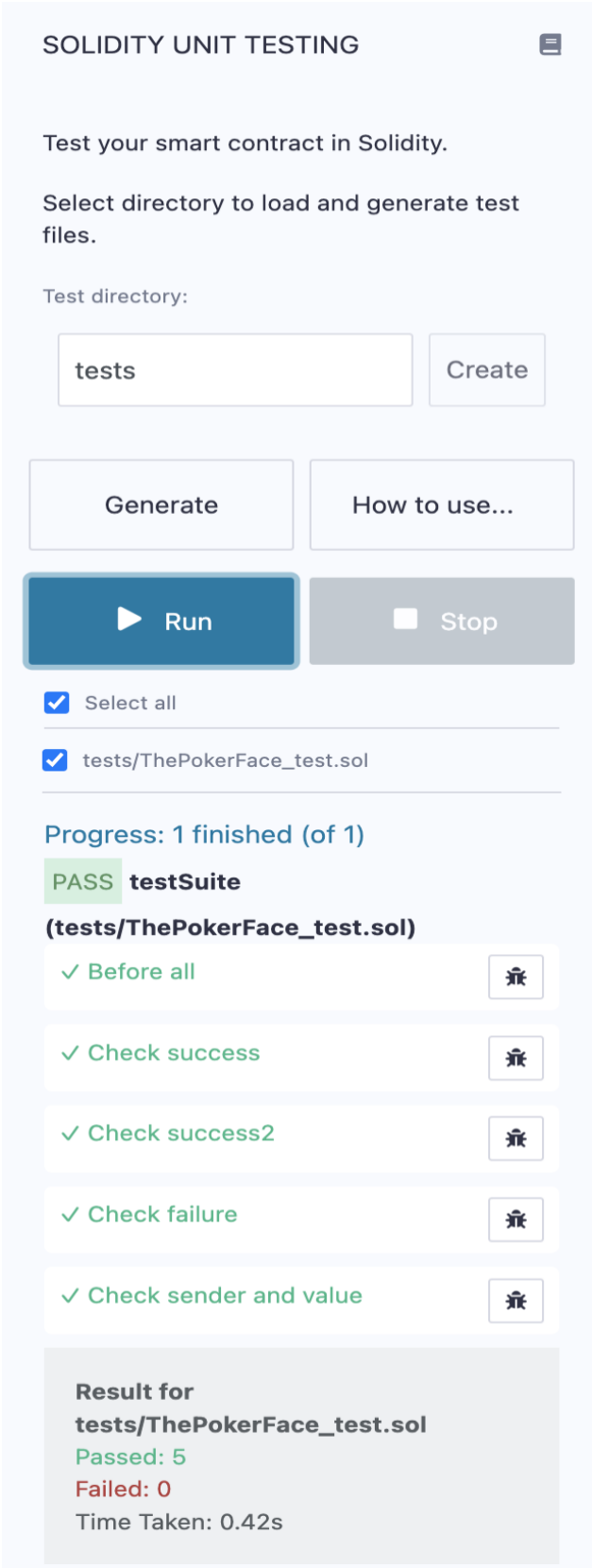


2- SOLIDITY STATIC ANALYSIS

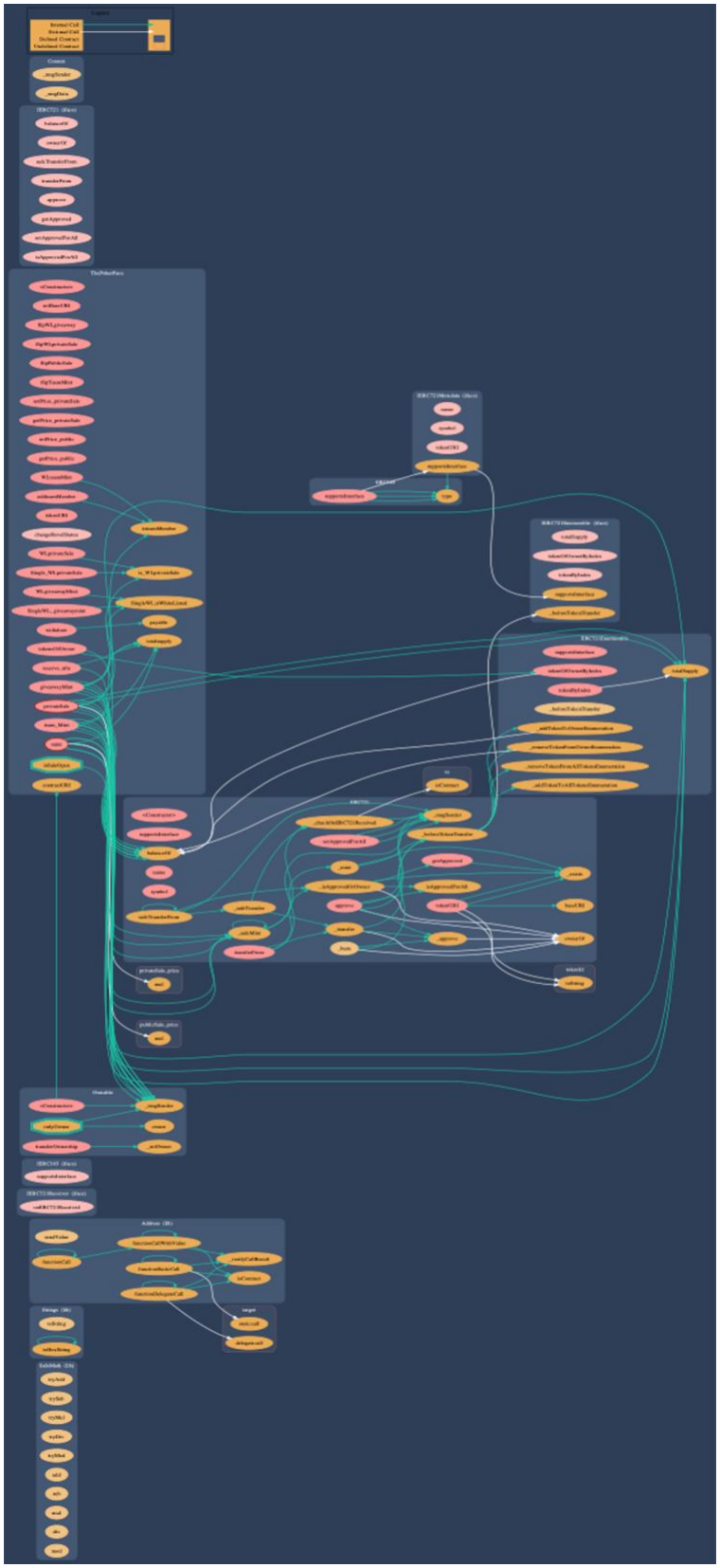
3- Inheritance graph



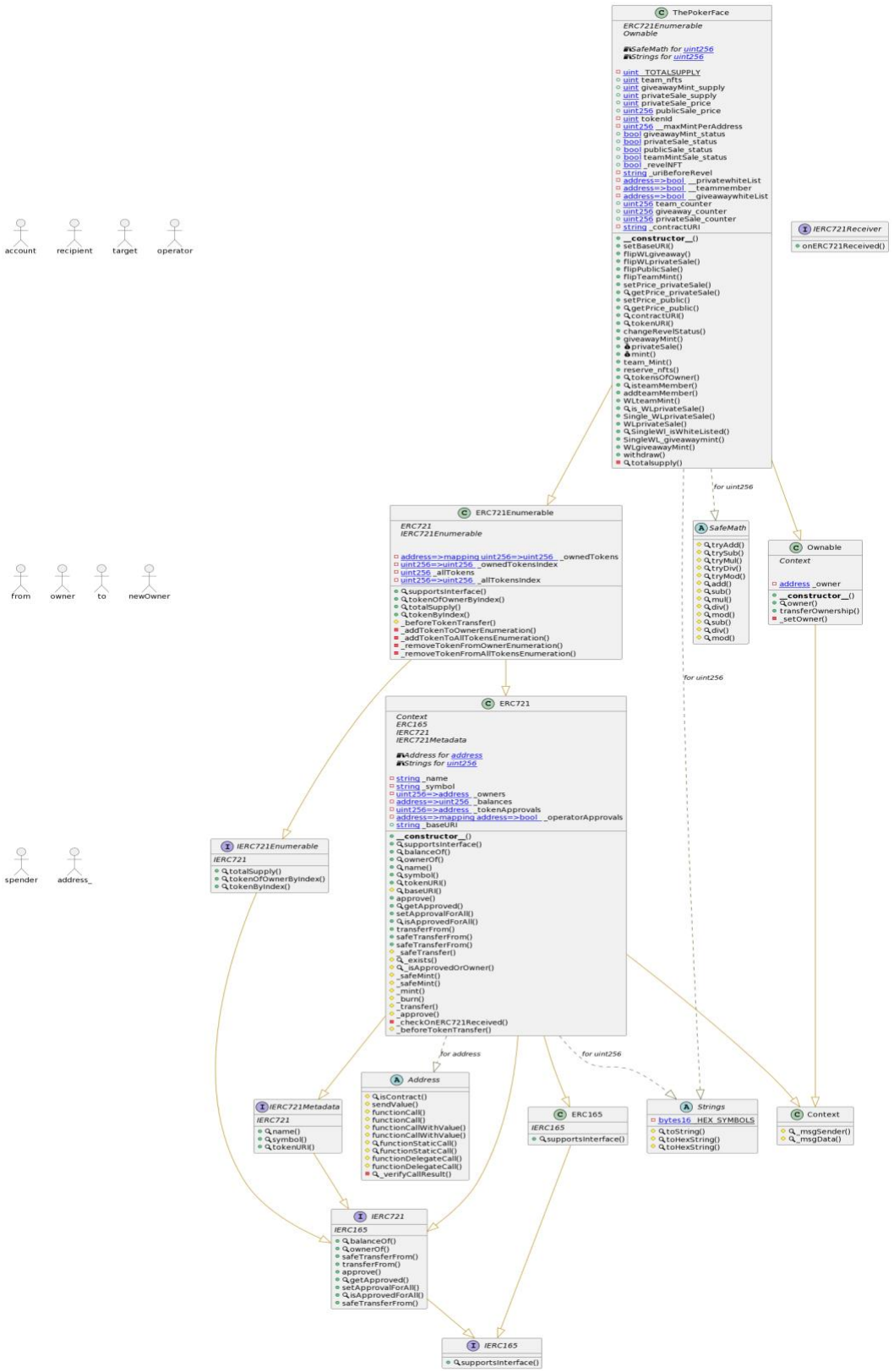
4- SOLIDITY UNIT TESTING



5- Call graph



Unified Modeling Language (UML)



Functions signature

```
Sighash   |   Function Signature
========================
16279055  =>  isContract(address)
88084605  =>  flipPublicSale()
884557bf  =>  tryAdd(uint256,uint256)
a29962b1  =>  trySub(uint256,uint256)
6281efa4  =>  tryMul(uint256,uint256)
736ecb18  =>  tryDiv(uint256,uint256)
38dc0867  =>  tryMod(uint256,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
f43f523a  =>  mod(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
b745d336  =>  div(uint256,uint256,string)
71af23e8  =>  mod(uint256,uint256,string)
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
c21d36f3  =>  functionStaticCall(address,bytes)
dbc40fb9  =>  functionStaticCall(address,bytes,string)
ee33b7e2  =>  functionDelegateCall(address,bytes)
57387df0  =>  functionDelegateCall(address,bytes,string)
18c2c6a2  =>  _verifyCallResult(bool,bytes,string)
150b7a02  =>  onERC721Received(address,address,uint256,bytes)
01ffc9a7  =>  supportsInterface(bytes4)
70a08231  =>  balanceOf(address)
6352211e  =>  ownerOf(uint256)
42842e0e  =>  safeTransferFrom(address,address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
095ea7b3  =>  approve(address,uint256)
081812fc  =>  getApproved(uint256)
a22cb465  =>  setApprovalForAll(address,bool)
e985e9c5  =>  isApprovedForAll(address,address)
b88d4fde  =>  safeTransferFrom(address,address,uint256,bytes)
18160ddd  =>  totalSupply()
2f745c59  =>  tokenOfOwnerByIndex(address,uint256)
4f6ccce7  =>  tokenByIndex(uint256)
06fdde03  =>  name()
95d89b41  =>  symbol()
c87b56dd  =>  tokenURI(uint256)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
8da5cb5b  =>  owner()
f2fde38b  =>  transferOwnership(address)
fc201122  =>  _setOwner(address)
6c0360eb  =>  baseURI()
```

```
24b6b8c0  =>  _safeTransfer(address,address,uint256,bytes)
f8e76cc0  =>  _exists(uint256)
4cdc9549  =>  _isApprovedOrOwner(address,uint256)
b3e1c718  =>  _safeMint(address,uint256)
6a4f832b  =>  _safeMint(address,uint256,bytes)
4e6ec247  =>  _mint(address,uint256)
9b1f9e74  =>  _burn(uint256)
30e0789e  =>  _transfer(address,address,uint256)
7b7d7225  =>  _approve(address,uint256)
1fd01de1  =>  _checkOnERC721Received(address,address,uint256,bytes)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
69025b5f  =>  _addTokenToOwnerEnumeration(address,uint256)
e03d890b  =>  _addTokenToAllTokensEnumeration(uint256)
68df0d53  =>  _removeTokenFromOwnerEnumeration(address,uint256)
4cbb4a0a  =>  _removeTokenFromAllTokensEnumeration(uint256)
55f804b3  =>  setBaseURI(string)
d4a91dc7  =>  flipWLgiveaway()
73a2e545  =>  flipWLprivateSale()
c4aa1a72  =>  flipTeamMint()
4d43613a  =>  setPrice_privateSale(uint256)
7b2aafb6  =>  getPrice_privateSale(uint256)
b449b5e7  =>  setPrice_public(uint256)
d02cedcd  =>  getPrice_public(uint256)
e8a3d485  =>  contractURI()
bd0a8439  =>  changeRevelStatus()
79ee8e3e  =>  giveawayMint(uint256)
b8d59a65  =>  privateSale(uint256)
a0712d68  =>  mint(uint256)
1c222814  =>  team_Mint(uint256)
20231c0a  =>  reserve_nfts(uint256)
8462151c  =>  tokensOfOwner(address)
4e0e9aa7  =>  isteamMember(address)
deeae065  =>  addteamMember(address)
8e683413  =>  WLteamMint(address[])
ddec7542  =>  is_WLprivateSale(address)
579206f4  =>  Single_WLprivateSale(address)
1eed5f5c  =>  WLprivateSale(address[])
16f5827e  =>  SingleWl_isWhiteListed(address)
ede829ee  =>  SingleWL_giveawaymint(address)
e8e6cd2e  =>  WLgiveawayMint(address[])
3ccfd60b  =>  withdraw()
72dd529b  =>  totalsupply()
```

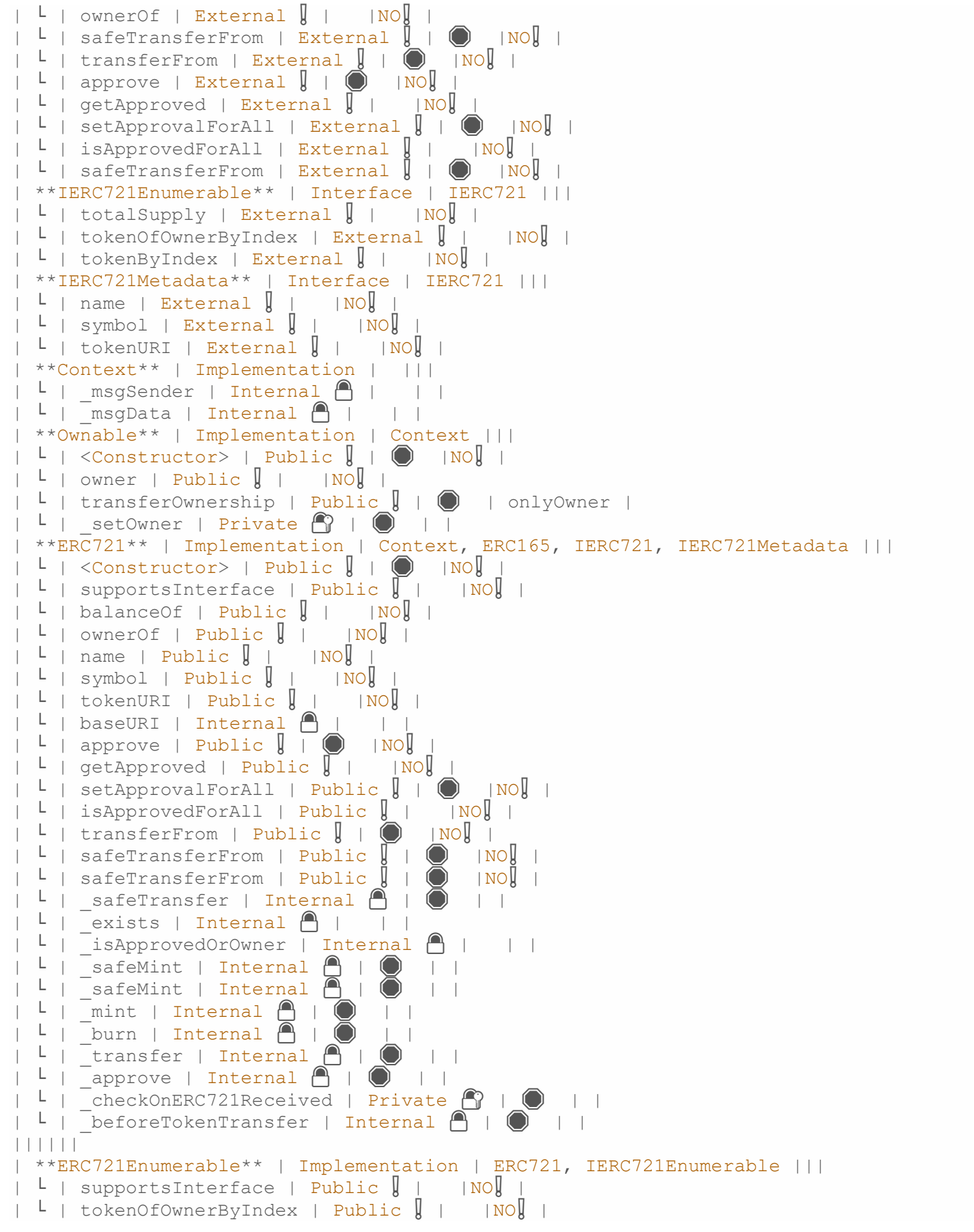
Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/ThePokerFace.sol	95bb51c1cf758c70937c58ee2d5f2d6b9983d0ed

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Strings	Library			
└	toString	Internal 🔒		
└	toHexString	Internal 🔒		
└	toHexString	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	_verifyCallResult	Private 🔐		
IERC721Receiver	Interface			
└	onERC721Received	External ❗	🛑	NO❗
IERC165	Interface			
└	supportsInterface	External ❗		NO❗
ERC165	Implementation	IERC165		
└	supportsInterface	Public ❗		NO❗
IERC721	Interface	IERC165		
└	balanceOf	External ❗		NO❗


└	ownerOf	External ❗		NO❗
└	safeTransferFrom	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	approve	External ❗	🛑	NO❗
└	getApproved	External ❗		NO❗
└	setApprovalForAll	External ❗	🛑	NO❗
└	isApprovedForAll	External ❗		NO❗
└	safeTransferFrom	External ❗	🛑	NO❗
IERC721Enumerable	Interface	IERC721		
└	totalSupply	External ❗		NO❗
└	tokenOfOwnerByIndex	External ❗		NO❗
└	tokenByIndex	External ❗		NO❗
IERC721Metadata	Interface	IERC721		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	tokenURI	External ❗		NO❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata		
└	<Constructor>	Public ❗	🛑	NO❗
└	supportsInterface	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	ownerOf	Public ❗		NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	tokenURI	Public ❗		NO❗
└	baseURI	Internal 🔒		
└	approve	Public ❗	🛑	NO❗
└	getApproved	Public ❗		NO❗
└	setApprovalForAll	Public ❗	🛑	NO❗
└	isApprovedForAll	Public ❗		NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	_safeTransfer	Internal 🔒	🛑	
└	_exists	Internal 🔒		
└	_isApprovedOrOwner	Internal 🔒		
└	_safeMint	Internal 🔒	🛑	
└	_safeMint	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_transfer	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_checkOnERC721Received	Private 🔐	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
ERC721Enumerable	Implementation	ERC721, IERC721Enumerable		
└	supportsInterface	Public ❗		NO❗
└	tokenOfOwnerByIndex	Public ❗		NO❗

└	totalSupply	Public ❗		NO❗
└	tokenByIndex	Public ❗		NO❗
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_addTokenToOwnerEnumeration	Private 🔐	🛑	
└	_addTokenToAllTokensEnumeration	Private 🔐	🛑	
└	_removeTokenFromOwnerEnumeration	Private 🔐	🛑	
└	_removeTokenFromAllTokensEnumeration	Private 🔐	🛑	
ThePokerFace	Implementation	ERC721Enumerable, Ownable		
└	<Constructor>	Public ❗	🛑	ERC721
└	setBaseURI	Public ❗	🛑	onlyOwner
└	flipWLgiveaway	Public ❗	🛑	onlyOwner
└	flipWLprivateSale	Public ❗	🛑	onlyOwner
└	flipPublicSale	Public ❗	🛑	onlyOwner
└	flipTeamMint	Public ❗	🛑	onlyOwner
└	setPrice_privateSale	Public ❗	🛑	onlyOwner
└	getPrice_privateSale	Public ❗		NO❗
└	setPrice_public	Public ❗	🛑	onlyOwner
└	getPrice_public	Public ❗		NO❗
└	contractURI	Public ❗		NO❗
└	tokenURI	Public ❗		NO❗
└	changeRevelStatus	External ❗	🛑	onlyOwner
└	giveawayMint	Public ❗	🛑	NO❗
└	privateSale	Public ❗	💸	NO❗
└	mint	Public ❗	💸	NO❗
└	team_Mint	Public ❗	🛑	NO❗
└	reserve_nfts	Public ❗	🛑	onlyOwner
└	tokensOfOwner	Public ❗		NO❗
└	isteamMember	Public ❗		NO❗
└	addteamMember	Public ❗	🛑	onlyOwner
└	WLteamMint	Public ❗	🛑	onlyOwner
└	is_WLprivateSale	Public ❗		NO❗
└	Single_WLprivateSale	Public ❗	🛑	onlyOwner
└	WLprivateSale	Public ❗	🛑	onlyOwner
└	SingleWl_isWhiteListed	Public ❗		NO❗
└	SingleWL_giveawaymint	Public ❗	🛑	onlyOwner
└	WLgiveawayMint	Public ❗	🛑	onlyOwner
└	withdraw	Public ❗	🛑	onlyOwner
└	totalsupply	Private 🔐		

Legend

Symbol	Meaning
🛑	Function can modify state
💸	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well-secured”.

✓ No volatile code.
✓ Not many high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.