TECHFUND Audit Score

Smart contract Audit Report

TCGC

Completed on 2022-03-17

Score **POSITIVE**

Risk level		
	Critical	0
	High	0
	Medium	0
	Low	1
	Note	0

Risk level detail

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

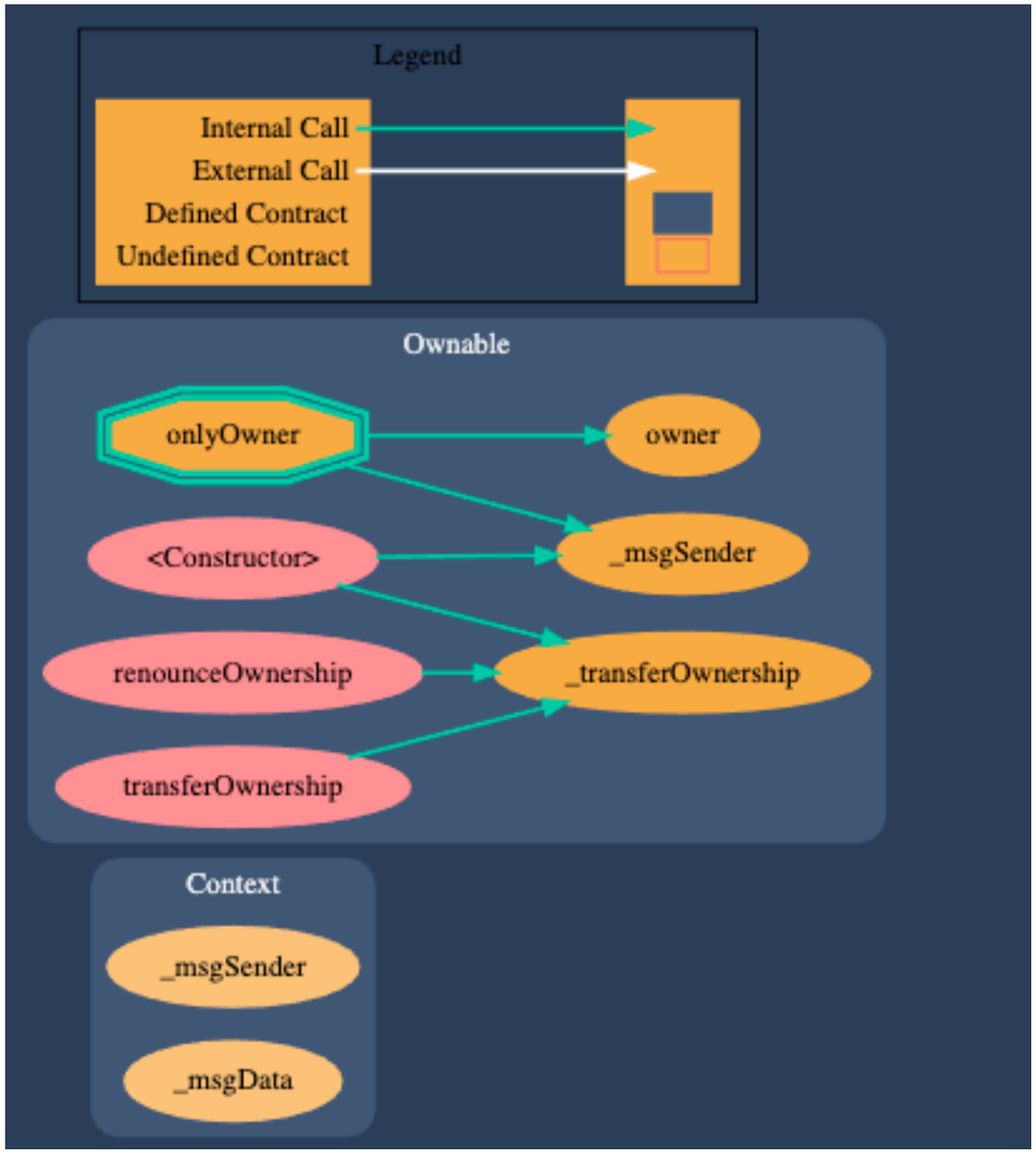
The tester arrives at the likelihood and impact estimates, they can now combine them to get a final severity rating for this risk. Note that if they have good business impact information, they should use that instead of the technical impact information.

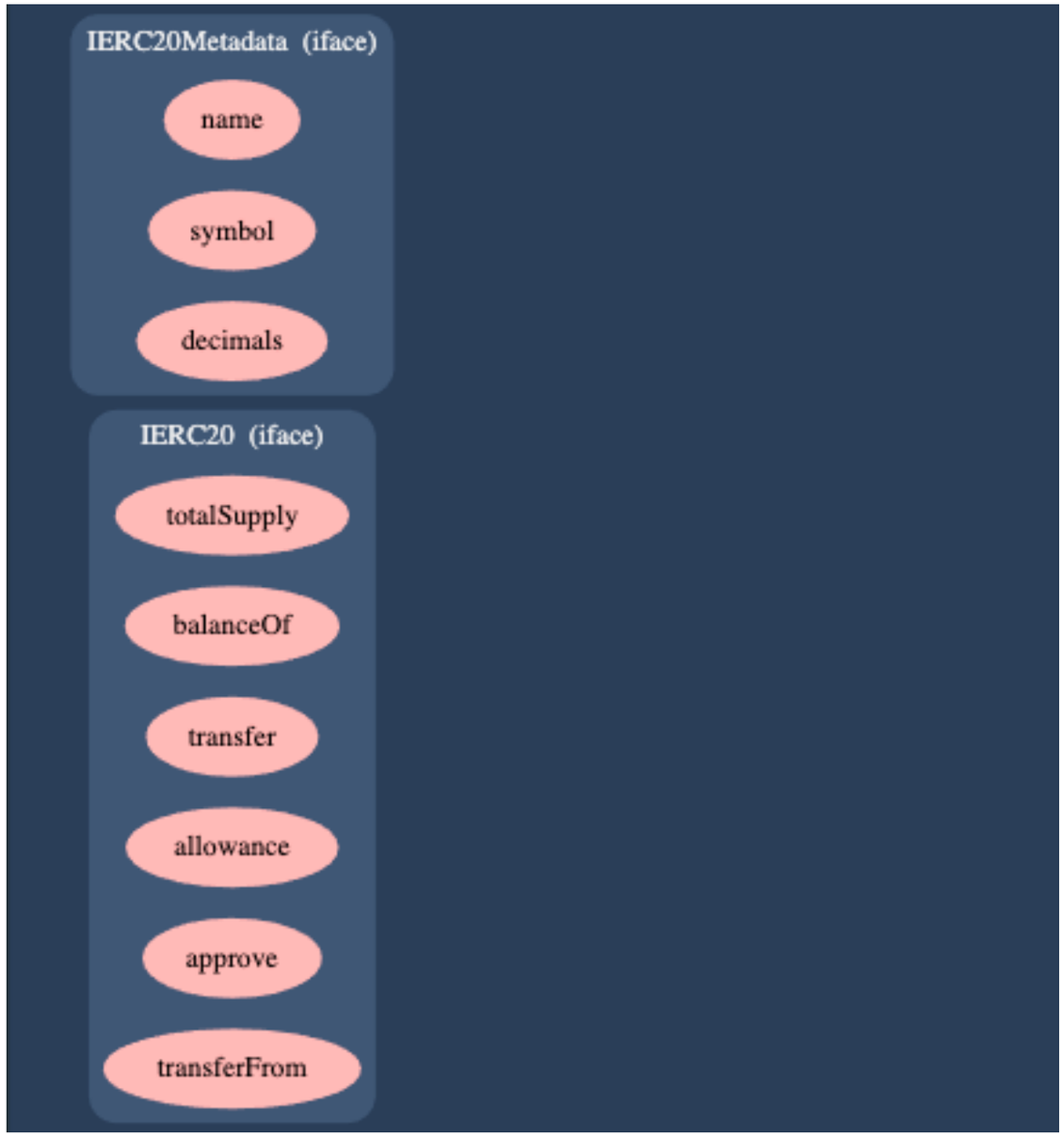
https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Vulnerability Review

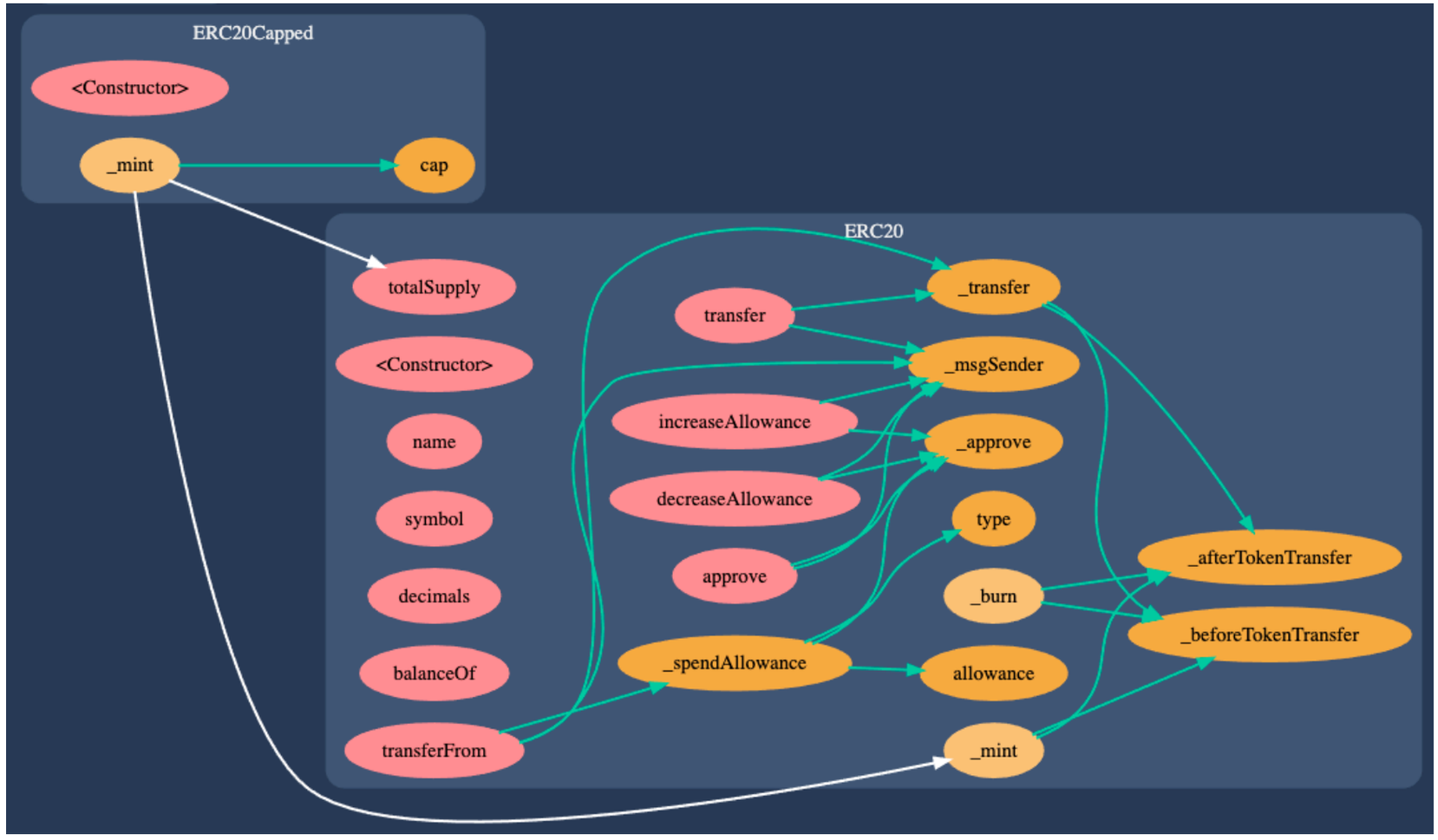
	Number of warnings
Compiler Version	0
State Variable Default Visibility	0
Arithmetic Operations	0
Mint Method	1
Integer Overflow / Underflow	0
Parity Multisig Bug	0
Callstack Depth Attack	0
Production Node modules security	0
Development Node modules security	0
Re-Entrancy	0
Double Withdrawal	0

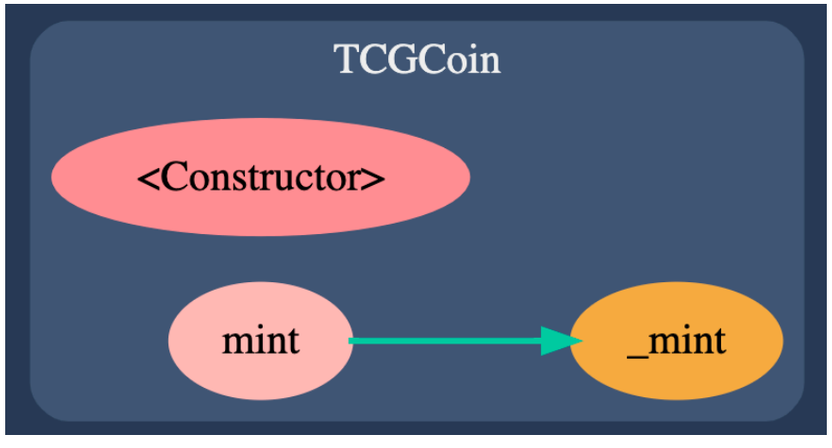
Call Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
owner
<Constructor>
_msgSender
renounceOwnership
_transferOwnership
transferOwnership
Context
_msgSender
_msgData


IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom

Call Graph : ERC20 Capped / TCGCoin





Mint Method 1

Although it is not a major issue in current case, but we highly suggest to use the function call by calling function using reference. Since both ERC20 class and ERC20Capped have references to _mint and it can create issues if not properly handled in future.

*Only ERC2Capped function has automatic capping check
require(ERC20.totalSupply() + amount <= cap(), "ERC20Capped: cap exceeded");

```
function mint(address to, uint256 amount) external onlyOwner {
    _mint(to, amount);
}
```

```
function mint(address to, uint256 amount) external onlyOwner {
    ERC20Capped._mint(to, amount);
}
```