Immutable Internal Contract Audit

Immutable ERC 20 Contracts

Internal Auditors	Peter Robinson
Assessment Date	April 11, 2024 and April 16, 2024
Final Report	April 16, 2024

Previous audits:

- March 28, 2024: Internal audit by Peter Robinson:
  https://github.com/immutable/contracts/blob/main/audits/token/202403-internal-audit-immutable-erc20.pdf
- List of all Open Zeppelin contracts audits:
  https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/audits
  Audit report for Open Zeppelin contracts v4.9:
  https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/audits/2023-05-v4.9.pdf

Contents

Description

Immutable supplies two ERC 20 contracts:

- ImmutableERC20MinterBurnerPermit.sol: This contract extends Open Zeppelin's ERC 20 Permit contract, providing minting by an minting administrator and burning by token owners. A maximum supply is specified when the contract is deployed. Additionally, it provides the concept of ownership, which is used to link deployed contracts to Immutable Hub accounts.
- ImmutableERC20FixedSupply.sol: This contract extends Open Zeppelin's ERC 20 contract, by creating a supply of tokens that is assigned to an owner address. Additionally, it provides the concept of ownership, which is used to link deployed contracts to Immutable Hub accounts.

Scope

Commit: aa6c1d43a4165a6e4d8cde302fe34b424b99bd32

Asset	Description
Smart Contracts	ERC20 contracts: • ImmutableERC20MinterBurnerPermit.sol • ImmutableERC20FixedSupplyNoBurn.sol
Threat model	https://github.com/immutable/contracts/blob/main/audits/token/202404-threat-model-preset-immutable-erc20.md

Team's Greatest Concerns

- The team would like a general review. They have no specific concerns.

Basic Contract Analysis

The table below analyzes[1] the contracts.

Type	File	Lines	nSLOC	Complexity
Contract	ImmutableERC20MinterBurnerPermit	71	26	29
Contract	ImmutableERC20FixedSupplyNoBurn	41	13	11

[1] Analysis produced using the Solidity Metrics VSCode extension.

Smart Contracts

ImmutableERC20MinterBurnerPermit

ImmutableERC20 extends Open Zeppelin's ERC20 Permit contract. It provides the following functionality:

- Games define a max supply for the token.
- The token contract must be owned by a non zero address at all times. Additional owners can be specified. The last owner can not renounce their ownership.
- Minting should be restricted to addresses that were granted the `minter` role.
- Contract is not upgradeable.

Roles:

- DEFAULT_ADMIN_ROLE: Role administrator for DEFAULT_ADMIN_ROLE, HUB_OWNER_ROLE and MINTER_ROLE
- MINTER_ROLE: Administrators that can mint more tokens.
- HUB_OWNER_ROLE: The contract is deemed to be related to an Immutable Hub account if the account has HUB_OWNER_ROLE.

External or Public Functions[2] that modify state:

Function Name	Function Selector	Authorisation Check	Notes
approve	095ea7b3	There is no authorisation check. However, the actions of the function only operate on balances related to msg.sender.	
burn	42966c68	There is no authorisation check. However, the actions of the function only operate on balances related to msg.sender.	
burnFrom	79cc6790	There is no authorisation check. However, the actions of the function only operate on allowances granted to msg.sender or balances related to msg.sender.	
decreaseAllowance	a457c2d7	There is no authorisation check. However, the actions of the function only operate on balances related to msg.sender.	
grantMinterRole	3dd1eb61	Msg.sender must have DEFAULT_ADMIN_ROLE	
grantRole	2f2ff15d	Msg.sender must be the role admin for the role	
increaseAllowance	39509351	There is no authorisation check. However, the actions of the function only operate on balances related to msg.sender.	
mint	40c10f19	Msg.sender must have MINTER_ROLE	
permit	d505accf	Msg.sender presents an EIP 712 data blob signed by the owner of the tokens to approve.	
renounceOwnership	715018a6	Always reverts	
renounceRole	36568abe	Msg.sender can only remove their own access.	

[2] The list of functions was determined using `forge inspect ImmutableERC20MinterBurnerPermit methods`

revokeMinterRole	69e2f0fb	Msg.sender must have DEFAULT_ADMIN_ROLE
revokeRole	d547741f	Msg.sender must be role admin for the role being revoked. For DEFAULT_ADMIN_ROLE, they must be another account with DEFAULT_ADMIN_ROLE.
transfer	a9059cbb	There is no authorisation check. However, the actions of the function only operate on balances related to msg.sender.
transferFrom	23b872dd	Msg.sender must have an allowance authorized by the token's owner.

External or Public Functions[3] that do not modify state:

Function Name	Function Selector	Notes
DEFAULT_ADMIN_ROLE	a217fddf	
DOMAIN_SEPARATOR	3644e515	
HUB_OWNER_ROLE	ba795d18	
MINTER_ROLE	d5391393	
allowance	dd62ed3e	
balanceOf	70a08231	
cap	355274ea	
decimals	313ce567	
eip712Domain	84b0196e	
getAdmins	31ae450b	
getRoleAdmin	248a9ca3	
getRoleMember	9010d07c	
getRoleMemberCount	ca15c873	
hasRole	91d14854	
maxSupply	d5abeb01	
name	06fdde03	
nonces	7ecebe00	
supportsInterface	01ffc9a7	
symbol	95d89b41	
totalSupply	18160ddd	

[3] The list of functions was determined using forge inspect ./<path>/<contract>.sol:<contract> methods

Upgradeable checks: This contract is not upgradeable.

Analysis of code logic: All issues related to internal reviews have been actioned. In particular, the following have been fixed:

- Documentation has been improved.
- Switched from ownable to HUB_OWNER_ROLE.
- Resolve access control issue with burn.

ImmutableERC20FixedSupplyNoBurn

ImmutableERC20FixedSupplyNoBurn extends Open Zeppelin's ERC20 and Ownable contracts. They offer the same functions, with the same authorisation checks, with the exception of renounceOwnership.

Roles: Owner

External or Public Functions[4] that modify state:

Function Name	Function Selector	Authorisation Check	Notes
approve	095ea7b3	There is no authorisation check. However, the actions of the functions only operate on allowances or balances related to msg.sender.	
decreaseAllowance	a457c2d7		
increaseAllowance	39509351		
renounceOwnership	715018a6	None	Function always reverts.
transfer	a9059cbb	There is no authorisation check. However, the actions of the functions only operate on allowances or balances related to msg.sender.	
transferFrom	23b872dd	Msg.sender must have an allowance authorized by the token's owner.	
transferOwnership	f2fde38b	Only owner	

External or Public Functions that do not modify state:

Function Name	Function Selector	Notes
allowance	dd62ed3e	
balanceOf	70a08231	
decimals	313ce567	
name	06fdde03	
owner	8da5cb5b	

[4] The list of functions was determined using `forge inspect ImmutableERC20FixedSupplyNoBurn methods`

symbol	95d89b41	
totalSupply	18160ddd	

Upgradeable checks: This contract is not upgradeable..

Analysis of code logic: The contract correctly creates a fixed supply of ERC-20 tokens, securely assigned to an owner upon deployment. The hub owner, token name and token symbol are also set.

Abuser Test Cases and Findings

Transfer Tokens to Incorrect Address

Classification: Informational

Description: An attacker could encourage game players that owned tokens to mistakenly transfer the tokens to an address not related to any contract or any private key. The tokens will be forever inaccessible. The attacker does not receive any benefit beyond the total circulating supply of tokens being reduced.

Action: A mitigation for this issue is for game players to be warned if they want to transfer assets to an address that is not a contract address and not an address with some IMX balance.

Status: Acknowledged by the team.