Password Store Report

Performed by: *0xShiki*

Table of Contents

Protocol Summary

Kitty Connect allows users to buy a cute cat from our branches and mint NFT for buying a cat. The NFT will be used to track the cat info and all related data for a particular cat corresponding to their token ids. Kitty Owner can also Bridge their NFT from one chain to another chain via Chainlink CCIP.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where I try to find as many vulnerabilities as possible. I can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

Risk Classification

		Impact		
		High	Medium	Low
Likelihood	High	H	H/M	M
	Medium	H/M	M	M/L
	Low	M	M/L	L

Audit Details

The findings described in this document correspond to the following github repo:

```
https://github.com/Cyfrin/2024-03-kitty-connect
```

Scope

```
- contracts/
  * KittyConnect.sol
  * KittyBridge.sol
```

Roles

- `Cat Owner`: User who buy the cat from our branches and mint NFT for buying a cat.
- `Shop Partner` - Shop partner provide services to the cat owner to buy cat.
- `KittyConnect Owner` - Owner of the contract who can transfer the ownership of the contract to another address.

Executive Summary

Issues found

Severtity	Number of issues found
High	1
Medium	0
Low	0
Gas	2
Info	0
Total	3

Findings

High

H-01. `KittyConnect::_updateOwnershipInfo` function doesn't update the ownership info of the kitty's previous owner, leads to confusion in managing and querying the ownership of NFTs.

Relevant GitHub Links

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

Vulnerability Details

When `KittyConnect::safeTransferFrom` function is called, it updates the ownership information of the NFT by calling `KittyConnect::_updateOwnershipInfo`. However, the function does not update the `KittyConnect::s_ownerToCatsTokenId` mapping, where the owner of the NFT is stored. ## Impact This could lead to confusion or inefficiencies in managing and querying the ownership of NFTs. ## Tools Used Manual Review

Proof of Code

The proof of concept is this test, which was already written:

PoC

```
function test_safetransferCatToNewOwner() public {
    string memory catImageIpfsHash = "ipfs://QmbxwGgBGrNdXPm84kqYskmcMT3jrzBN8LzQjixvkz4c62";
    uint256 tokenId = kittyConnect.getTokenCounter();
    address newOwner = makeAddr("newOwner");

    vm.prank(partnerA);
    kittyConnect.mintCatToNewOwner(user, catImageIpfsHash, "Meowdy", "Ragdoll", block.timestamp);

    vm.prank(user);
    kittyConnect.approve(newOwner, tokenId);

    vm.expectEmit(false, false, false, true, address(kittyConnect));
    emit CatTransferredToNewOwner(user, newOwner, tokenId);
    vm.prank(partnerA);
    kittyConnect.safeTransferFrom(user, newOwner, tokenId);

    assertEq(kittyConnect.ownerOf(tokenId), newOwner);
    assertEq(kittyConnect.getCatsTokenIdOwnedBy(user).length, 0);
    assertEq(kittyConnect.getCatsTokenIdOwnedBy(newOwner).length, 1);
    assertEq(kittyConnect.getCatsTokenIdOwnedBy(newOwner)[0], tokenId);
    assertEq(kittyConnect.getCatInfo(tokenId).prevOwner[0], user);
    assertEq(kittyConnect.getCatInfo(tokenId).prevOwner.length, 1);
    assertEq(kittyConnect.getCatInfo(tokenId).idx, 0);
}
```

By running this test, we can see that this assertion fails:

```
assertEq(kittyConnect.getCatsTokenIdOwnedBy(user).length, 0);
```

Recommendations

To prevent this, we can remove the tokenId from the array of the previous owner in `KittyConnect::s_ownerToCatsTokenId` mapping. This can be done by adding the following line of code in the `KittyConnect::_updateOwnershipInfo` function:

```
function _updateOwnershipInfo(
    address currCatOwner,
    address newOwner,
    uint256 tokenId
) internal {
+   // Get the index of the token ID in the array
+   uint256 tokenIndex = s_ownerToCatsTokenId[currCatOwner].length;
+   for (uint256 i = 0; i < tokenIndex; i++) {
+       if (s_ownerToCatsTokenId[currCatOwner][i] == tokenId) {
+           tokenIndex = i;
+           break;
+       }
+   }
+
+   // Swap the token ID to be removed with the last element in the
   array
+   s_ownerToCatsTokenId[currCatOwner][tokenIndex] =
   s_ownerToCatsTokenId[
+        currCatOwner
+   ][s_ownerToCatsTokenId[currCatOwner].length - 1];

+   // Pop the last element from the array
+   s_ownerToCatsTokenId[currCatOwner].pop();
    s_catInfo[tokenId].prevOwner.push(currCatOwner);
    s_catInfo[tokenId].idx = s_ownerToCatsTokenId[newOwner].length;
    s_ownerToCatsTokenId[newOwner].push(tokenId);
}
```

Gas

G-01. Using `require` instead of custom errors, leads to gas inefficiency.

Relevant GitHub Links

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

Description

All of the error handling in the `KittyConnect` contract is done using the require function. Since Solidity v0.8.4, custom reverts were presented, which are more gas efficient than using `require`.

Impact

Leads to spending more gas than necessary.

Tools Used

Manual Review

Recommendations

Read about custom errors in this solidity blog, and rewrite them with the new syntax.

G-02. Reading from storage in `KittyConnect::constructor` every iteration in the for-loop, which is gas inefficient.

Relevant GitHub Links

https://github.com/Cyfrin/2024-03-kitty-connect/blob/c0a6f2bb5c853d7a470eb684e1954dba261fb167/src/KittyConne

Vulnerability Details

In `KittyConnect::constructor`, the function reads from storage in every iteration of the for-loop. Reading from storage is more expensive than reading from memory.

Impact

Leads to spending more gas than necessary.

Tools Used

Manual Review

Recommendations

Consider storing the length of the array in a local variable and using it in the for-loop. This will reduce the gas cost of the function.

```
constructor(
    address[] memory initShops,
    address router,
    address link
) ERC721("KittyConnect", "KC") {
+   uint256 initialShopsArray = initShops.length;
+   for (uint256 i = 0; i < initialShopsArray; i++) {
-   for (uint256 i = 0; i < initShops.length; i++) {
        s_kittyShops.push(initShops[i]);
        s_isKittyShop[initShops[i]] = true;
    }

    i_kittyConnectOwner = msg.sender;
    i_kittyBridge = new KittyBridge(router, link, msg.sender);
}
```