**Security Audit Report**

**By SAPNA**

1. **Contract Overview:** This contract is allows us to store and retrieve data user’s amount and information and contract owner’s address, which demonstrates the use of struct, mapping and storage references.
2. **Code with Errors:**

pragma solidity ^0.4.23;

contract StorageVictim {

address owner;

struct Storage {

address user;

uint amount;

}

mapping(address => Storage) storages;

function StorageVictim() public {

owner = msg.sender;

}

function store(uint \_amount) public {

// uninitialised pointer. str.user points to the storage address 0 which is "owner"

Storage str;

str.user = msg.sender;

str.amount = \_amount;

storages[msg.sender] = str;

}

function getStore() public view returns (address, uint) {

Storage str = storages[msg.sender];

return (str.user, str.amount);

}

function getOwner() public view returns (address) {

return owner;

}

}

1. **Corrected Code:**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0; //corrected license

contract StorageVictim {

address owner; //stores owner's address

struct Storage { //Defines a new data type 'Storage' as a struct

address user; //stores user's address

uint amount; //stores amount of user

}

mapping(address => Storage) storages; //mapping to store 'Storage' structs for different addresses

constructor() { //constructor created instead of function

owner = msg.sender; //sets owner to address

}

function store(uint \_amount) public { //arguments given and function to store user's amount

Storage storage str = storages[msg.sender]; //reference to the storage location of the 'Storage'

str.user = msg.sender; //sets user's field to caller's address

str.amount = \_amount; //sets amount field to provided amount

}

function getStore() public view returns (address, uint) { //function that retrieves user's amount

Storage storage str = storages[msg.sender]; //creates storage reference for caller's address

return (str.user, str.amount); //returns user and amount to 'Storage'

}

function getOwner() public view returns (address) { //function to retrieve owner's address

return owner; //returns the address

}

}