Metacrafters Smart Contract Audit Report

Contract Name: StorageVictim
Version: 0.4.23
Audit Performed By: Sabari H

Findings:

Uninitialized Pointer Vulnerability

Vulnerability: Severe

The Storage pointer `str` is uninitialized. Due to this `str.user` points to address `` `0` `` by default which is the contract owner's address.

Recommended Change

Initialize the `str` to `storages[msg.sender]` in the store function

Mutable State

Vulnerability: Moderate

The state variable which stores the address of the owner is left mutable. Since unnecessary updations to the owner might cause issues, it is recommended to make the state immutable

Recommended Change

Change the owner state declaration to `` `address immutable owner;` ``

Constructor Syntax

Vulnerability: Normal

The syntax of the constructor has been changed from the contract name to `constructor`

Recommended Change

Change the constructor signature from `` `function StorageVictim() public {...}` `` to `` `constructor() {...}` ``

Minor Changes

1. **Uint to Uint256:** Use `uint256` for future-proof code. `uint256` provides better information about the size and adds clarity to the code. Also the size of uint might change in the further updates
2. **SPDX License:** The license line could be added. The License comment line is a standard way to specify the license under which the contract is released.
3. **Parameter naming:** The parameter ``` `_amount` ``` of the function store could be changed to ``` `amount` ```. This naming convention is recommended by the solidity team.

Additional Comments

- Consider using the latest version of solidity for security enhancements
- Consider adding comments to make it more readable

Conclusion

The contract “StorageVictim” contains a critical vulnerability related to uninitialized pointers. The recommended update might be helpful in enhancing the security of the contract.

Disclaimer:

This audit report might not contain all the bugs. So it is advised to perform further testing before deploying the contract to production.