Metacrafters Smart Contract Audit Report

Contract Name: StorageVictim **Version: 0.4.23**
Audit Performed By: Michael Dean **Date: 26/07/2024**
No. of contracts: 1 **No. of Functions: 4**

Findings

VULNERABILITY: CRITICAL

i. Uninitialized Storage Variables Vulnerability:

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 `Storage memory str;` in the store function

POC:

```
function store(uint _amount)  public  { Storage str; str.user =  msg.sender; str.amount = _amount; storages[msg.sender]  = str; }
```

VULNERABILITY: MEDIUM

ii. Outdated solidity compiler:

The contract uses an outdated version of solidity which might introduce certain vulnerabilities and would not be compatible with recent versions of solidity compiler

Recommended Change:

Change the solidity compiler version to a more recent version.

POC:

```
pragma  solidity  0.4.23;
```

VULNERABILITY: INFORMATIONAL

iii. Deprecated Constructor Syntax

Defining constructors as functions with the same name as the contract is deprecated.

Recommended Change:

Use the constructor keyword instead.

VULNERABILITY: INFORMATIONAL

iv. Missing SPDX-License-Identifier

There is no definition of a license identifier, which might flag as an error in certain development environment.

Recommended Change:

Add a specified License identifier, you could use `unlicensed` or a specific identifier.

VULNERABILITY: INFORMATIONAL

v. Address owner can be marked `immutable`:

Since the address of the owner is designed to be assigned only once at construction, gas could be saved at deployment by marking the owner address variable as `immutable`.

Recommended Change:

State variable `owner` should be marked as immutable.

POC:

```
address owner;
```

Summary

The contract “StorageVictim” contains 1 critical vulnerability, 1 medium vulnerability and 3 informational vulnerability. 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.