Smart Contracts Exercise 04: Unbreakable Vault

1 Introduction

In this exercise, you will be tasked with breaching several vaults, one by one. You will gain familiarity with the JavaScript library Ethers.js, which is designed to facilitate interaction with the Ethereum blockchain and its ecosystem. We will also demonstrate how to work in Remix IDE, an open-source development environment accessible through a web browser. Additionally, you will learn about blockchain data transparency, the differences between `msg.sender` and `tx.origin`, and how to predict `blockhash` or `block.timestamp` in certain scenarios.

Prerequisites

Ensure that you have already installed the following on your system:

- **Node.js** - `https://nodejs.org/en/` An open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
- **NPM**: Node Package Manager, which comes with Node.js.

Open your terminal and run the following commands to verify the installations:

```
$ node -v
$ npm -v
```

Both commands should return the installed version numbers of Node.js and NPM respectively. Node.js provides the runtime environment required to execute JavaScript-based tools like Hardhat, while NPM is used to manage the packages and dependencies needed for development. It is recommended that you use NPM 7 or higher.

For the purposes of this exercise, you will need an Infura API Key and a configured wallet. If you do not have this set up yet, we recommend going through the Smart Contracts Exercise 01: Hello, Blockchain World! where everything is explained. You should have configuration variables set for Hardhat projects. You can verify this by running:

```
$ npx hardhat vars get INFURA_API_KEY
$ npx hardhat vars get SEPOLIA_PRIVATE_KEY
```

Project Set Up

To get started, visit the following GitLab repository and clone it to your local machine. This repository contains a template in which you will complete this exercise. After you clone the repository, start with the following command within your project folder:

```
$ npm install
```

This will install all the necessary dependencies for the project. Your implementation will be in the `contracts` and `test` folders. There will be multiple vaults in this exercise that you need to breach, each one having a separate test. To see if you have completed the task successfully, run `npm run vaultXX` where `XX` is the number of the vault you are trying to breach. For example, to test the first vault, run:

```
$ npm run vault01
```

To run all tests at once, run:

```
$ npx hardhat test
```