Разработка безопасных децентрализованных приложений на блокчейне Ethereum

Код: DEV-039

Длительность: 24 ч.

Описание:

Современные блокчейн-платформы позволяют создавать распределенные отказоустойчивые приложения (dApps), способные накапливать и распределять цифровые активы пользователей по описанным в смарт-контракте правилам. Краеугольным камнем в разработке dApp является безопасность приложения, так как оно оперирует ценными активами и криптовалютой пользователей.

Многочисленные инциденты по взлому смарт-контрактов (TheDAO, Parity Wallet, etc) убедительно показали, что разработчики не умеют строить надёжных распределенных приложений и нуждаются в использовании специальных средств, повышающих надёжность разрабатываемых ими программ.

На курсе рассматривается процесс безопасной разработки dApp для блокчейн платформы Ethereum и языка смарт-контрактов Solidity, с фокусом на вопросах надёжности и безопасности разрабатываемого решения через использование современных средств тестирования и статического анализа смарт-контрактов: Truffle, Mythril, SolTracer.

За время курса будет рассмотрен пример полноценного распределенного приложения для отслеживания цепочки поставок ценных изделий на базе блокчейна Ethereum, с подробным аудитом безопасности используемых смарт-контрактов.

Курс включает в себя несколько практических заданий, направленных на закрепление теоретического материала.

Цели:

Цели курса:

- Сформировать стойкое понимание устройства современных распределенных приложений блокчейна;
- Сформировать представление о возможностях смарт-контрактов в блокчейне Ethereum и некоторых типовых уязвимостях этого вида приложений;
- Познакомить с языком программирования Solidity и средой разработки Remix IDE;
- Выработать первичные навыки использования современных инструментов повышения надёжности смарт-контрактов Truffle, Mythril, SolTracer.

Разбираемые темы:

1. Обзор архитектуры блокчейн-платформы Ethereum (1 ч.)
2. Построение архитектуры типичного dApp: front-end (GUI), смарт-контракт на блокчейне, кошельки (2 ч.)
3. Модель исполнения смарт-контрактов: как они устроены и как взаимодействуют с окружающим миром (1 ч.)
4. Язык программирования Solidity и среда прототипирования смарт-контрактов RemixIDE (3 ч.)
5. Устройство пользовательского интерфейса dApp (5 ч)
6. Написание тестовых сценариев смарт-контратков в Truffle Framework (4 ч.)
7. Статический анализ смарт-контрактов используя Mythril. (4 ч.)
8. Автоматическое тестирование смарт-контрактов в SolTracer (4 ч.)

Целевая аудитория:

Разработчики, тестировщики, архитекторы, аудиторы ПО, технические консультанты по блокчейн-проектам

Предварительная подготовка – общее:

Знание одного из современных языков программирования (C++/C#/Java/JavaScript/Python/...) Минимальные навыки работы в командной строке Linux