I. BLOCKCHAIN

Blochain là công nghệ chuỗi – khối, cho phép truyền tải dữ liệu một cách an toàn dựa trên hệ thống mã hóa phức tạp

Trong lĩnh vực cryptocurrency thì blockchain là một sổ cái kỹ thuật số phân tán, phi tập trung

Cách thức hoạt động của blockchain là:

- Mỗi khối (block) chứa thông tin dữ liệu giao dịch và được liên kết với khối trước đó.
- Khi data đã được cập nhật vào blockchain và được chấp nhận thì khó có thể sửa đổi, nếu sửa sẽ để dấu vết

⇒ Blockchain được thiết kế để chống lại việc gian lận, thay đổi xóa sửa của dữ liệu.

II. ETHEREUM

- Và nền tảng điện toán ứng dụng công nghệ Blockchain mà nhóm em nghiên cứu là Ethereum.
- Ethereum là một nền tảng điện toán có tính chất phân tán, công cộng, mã nguồn mở dựa trên công nghệ Blockchain.
- Có tính năng hợp đồng thông minh, tạo thuận lợi cho các thỏa thuận hợp đồng trực tuyến.
- Cung cấp một loại tiền mã hóa gọi là "Ether", có thể được chuyển giữa các tài khoản và được sử dụng để trả công cho các thợ đào giúp thực hiện việc tính toán.

1. Khác nhau so với Bitcoin?

Bitcoin	Ethereum
Một loại tiền tệ và để lưu giữ giá trị (Bitcoin cũng có thể xử lý được hợp đồng thông minh)	Một nền tảng giao dịch hợp đồng thông minh phân tán (Ethereum cũng có thể được sử dụng như một loại tiền tệ)
Dùng để thanh toán hàng hóa và dịch vụ tại bất cứ nơi nào đồng tiền này được chấp nhận	- Đồng tiền Ether của mạng lưới Ethereum không được thiết kế như một giải pháp thanh toán thay thế - Dùng để thúc đẩy các lập trình viên và các tổ chức vận hành các ứng dụng phi tập trung trong mạng Ethereum.
Thời gian tạo khối là 10 phút	Thời gian tạo khối là 14-15 giây
	Dùng giao thức GHOST (Giúp nhanh hơn ; Chống lại việc đào mỏ tập trung)
Số lượng giới hạn ở 21 triệu với phần thưởng giảm còn 1 nửa sau mỗi 4 năm	- Không giới hạn số lượng ether - Lượng lạm phát ether hàng năm không xđ rõ
Phí giao dịch Bitcoin bị cạnh tranh trực tiếp với nhau để vào được khối của Bitcoin mà bị giới hạn	Phí giao dịch được trả bằng Gas (quy đổi được ra ether), tính dựa trên khối lượng tính toán, băng thông, lưu trữ
Dùng ASIC	- Chống lại ASIC. - Người đào Ethereum phải sử dụng card đồ họa vì hàm băm của Ethereum yêu cầu sử dụng bộ nhớ.
	- Cho phép chạy mã Turing-complete - Cho phép mọi tính toán được thực thi nếu có đủ khả năng tính toán và thời gian. - Nhiều rủi ro tấn công hơn so với cấu trúc đơn giản của Bitcoin

2. Hợp đồng thông minh là gì?

- Là cơ chế trao đổi xác định, được kiểm soát bởi các phương tiện kỹ thuật số, cho phép tạo ra các giao thực không cần dựa trên sự tin cậy.

- Là bộ giao thức đặc biệt dựa trên Blockchain, có khả năng tự đưa ra các điều khoản và thực thi thỏa thuận giữa các bên trong hợp đồng thông minh, cho phép mọi người triển khai giao dịch mà không cần thông qua bên thứ ba.

- Những giao dịch này hoàn toàn dễ dàng truy dấu và không thể bị can thiệp hoặc đảo chiều. Smart Contract chứa toàn bộ những thông tin chi tiết về các điều khoản và thực hiện chúng một cách tự động.

3. Hợp đồng thông minh thực chất là gì?

- Là các đoạn mã BYTECODE được viết và dịch bằng ngôn ngữ Solidity. Đoạn mã này sau đó được chuyển tới tất cả các Node của Blockchain và sẽ thực thi hợp đồng trên máy ảo EVM (trong trường hợp của Ethereum). Chính việc Hợp đồng thông minh là các đoạn mã nên rất dễ có những lỗ hổng tồn tại dẫn đến việc bị Hacker kiểm soát.

III. REENTRANCY

1. Lỗ hổng Reentrancy là gì?

- Là 1 lỗi trong Solidity, gây rất tốn kém do ảnh hưởng đến dữ liệu và tài sản trong hợp đồng thông minh.

- Reentrancy - khai thác gửi đệ quy:

* Tấn công Reentrancy xảy ra khi kẻ tấn công rút tiền từ mục tiêu bằng cách gọi đệ quy chức năng rút tiền (như trường hợp của DAO).

* Khi contract không cập nhật trạng thái (số dư người dùng) trước khi gửi tiền, kẻ tấn công có thể liên tục gọi chức năng rút tiền để rút hết tiền trong contract.

* Bất cứ khi nào kẻ tấn công nhận Ether, contract của hắn tự động gọi fallback function (fallback function là một hàm không tên trong contract, được thực thi khi hàm gọi đến không trùng với bất cứ hàm nào trong contract, hoặc hàm gửi ether mà không chứa dữ liệu - nơi lại gọi hàm rút tiền một lần nữa).

* Lúc này cuộc tấn công đi vào vòng lặp đệ quy và contract sẽ không thể cập nhật số dư của kẻ tấn công.

- DAO là một tổ chức được thiết kế để tự động hóa và phi tập trung. Mục tiêu của nó là mã hóa các quy tắc và bộ máy ra quyết định của một tổ chức, loại bỏ sự cần thiết của các tài liệu và con người trong quản lý, tạo ra một cấu trúc với sự kiểm soát phi tập trung.

2. Tấn công Reentrancy

- Một trong những mối nguy hiểm lớn của việc gọi các hợp đồng bên ngoài (external contract) là chúng có thể chiếm quyền điều khiển và thực hiện các thay đổi đối với dữ liệu của bạn theo cách không mong đợi.

- Sự sụp đổ của DAO (Decentralized Autonomous Organization) đều là các lỗi thuộc loại này. Cụ thể, trong ngày 17/6/2016, DAO bị tin tặc chiếm dụng 3.6 triệu Ether (tương đương 50 triệu đô la Mỹ) bằng cách khai thác lỗ hổng Reentrancy.

3. Các dạng chính của Reentrancy

3.1. Reentrancy xảy ra trên một hàm đơn lẻ (Reentrancy on a Single Function)

- Một hàm được gọi lặp đi lặp lại trước khi lời gọi hàm đầu tiên của nó hoàn tất. Theo cách này, các lời gọi hàm liên tiếp sẽ phá vỡ tính toàn vẹn của dữ liệu nếu không được kiểm soát tốt.

- Số dư tài khoản người dùng KHÔNG được thiết lập giá trị về 0 cho đến khi lời gọi hàm đầu tiên hoàn tất. Do đó, các lời gọi gọi thứ 2, thứ 3... sẽ thực hiện rút tiền thành công ra khỏi tài khoản mà vẫn không chịu sự giới hạn nào.

- Lỗi Reentrancy xảy ra do cơ chế hoạt động của hàm Call và Fallback trong quá trình thực hiện lời gọi hàm từ bên ngoài, hay trong trường hợp một Hợp đồng nhận được Ether.

3.2. Reentrancy xảy ra liên hàm (Cross-function Reentrancy)

- Kẻ tấn công cũng có thể thực hiện một cuộc tấn công tương tự bằng cách sử dụng hai hàm khác nhau có cùng trạng thái.

- Kẻ tấn công gọi hàm **transfer ()** khi mã nguồn của chúng đang được thực thi trên lời gọi hàm bên ngoài (external call) thông qua phương thức **withdrawBalance ()**. Vì số dư của kẻ tấn công chưa được đặt thành 0, nên hắn có thể tiếp tục chuyển tiền về tài khoản của hắn bằng lời gọi hàm **transfer ()** mặc dù hắn đã nhận được khoản rút tiền thông qua hàm **withdrawBalance ()**. Lỗ hổng này cũng được sử dụng trong cuộc tấn công **DAO**.

- Lỗi Reentrancy cũng có thể xảy ra trên nhiều hợp đồng, nếu các hợp đồng đó chia sẻ trạng thái.

RESEARCH METHOD

1. ARTIFICIAL NEURAL NETWORK (ANN)

1.1. Sơ lược

Gồm 3 thành phần chính:

- Input layer: gồm 1 layer
- Output layer: gồm 1 layer
- Hidden layer: có thể có 1 hay nhiều layer tùy vào bài toán cụ thể.

ANN hoạt động theo hướng mô tả lại cách hoạt động của hệ thần kinh với các neuron được kết nối với nhau

- Trong ANN, trừ **input layer** thì tất cả các node thuộc các layer khác đều full-connected với các node thuộc layer trước nó.

- Mỗi node thuộc **hidden layer** nhận đầu vào từ layer trước và kết hợp với trọng số để ra được kết quả.

2. STANDARD RNN (Recurrent Neural Network)

2.1. Định nghĩa

- Có tính chất nhớ

- Đầu vào và đầu ra của mạng neuron này độc lập, không liên kết thành chuỗi với nhau.

- Mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ... vì những dự đoán tiếp theo như từ tiếp theo phụ thuộc vào vị trí của nó trong câu và những từ đằng trước nó.

➔ RNN ra đời với ý tưởng chính là:

- Sử dụng chuỗi các thông tin
- Sử dụng một bộ nhớ để lưu lại thông tin từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại.

2.2. Ứng dụng

- Thành công nhất ở lĩnh vực xử lý ngôn ngữ tự nhiên. RNN cho phép ta dự đoán xác suất của một từ mới nhờ vào các từ đã biết liền trước nó.

- Trên lý thuyết thì RNN có khả năng nhớ được những tính toán (thông tin) ở trước nó, nhưng mô hình **RNN truyền thống** không thể nhớ được những bước ở xa do bị mất mát đạo hàm

➔ Nên những thành công của mô hình này chủ yếu đến từ một mô hình cải tiến khác là **LSTM** (Long Short-Term Memory). LSTM về cơ bản cũng giống với RNN truyền thống ngoài việc thêm các cổng tính toán ở **hidden layer** để quyết định giữ lại các thông tin nào.

3. LSTM

3.1. Định nghĩa

- Là dạng đặc biệt của RNN – có khả năng học được các phụ thuộc xa.

- Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

3.2. Nhược điểm

- Khi học xuôi, LSTM nhận diện được đoạn code / văn bản đó (theo chiều forward)

➔ Phải kết hợp với Bidirectional để học ngược xuôi --> Bi - LSTM

4. BIDIRECTIONAL LSTM

5. ATTENTION