Motivation:

Ethereum là nền tảng điện toán phân tán, dựa trên công nghệ chuỗi khối (Blockchain) có khả năng thực thi hợp đồng thông minh (Smart Contract) - tức là điều khoản được ghi trong hợp đồng sẽ được thực thi một cách tự động khi các điều kiện trước đó được thỏa mãn, không ai có thể can thiệp vào. Smart Contract được viết bằng các ngôn ngữ lập trình dễ xảy ra lỗi như Solidity, … do đó trong Smart contract thường tồn tại các lỗ hổng có thể bị attacker khai thác. Reentrancy (Hình 1) là lỗ hổng phổ biến trong Smart Contract (nằm trong danh sách top 10 DASP). Một attacker đánh cắp 60 triệu đô la Mỹ bằng cách sử dụng lỗ hổng này trong các tổ chức tự trị phi tập trung (DAO) (https://www.coindesk.com/understanding-dao-hack-journalists).

Sử dụng mô hình học sâu (deep learning) và cơ chế attention nhằm mục đích cải thiện và nâng cao khả năng tự động phát hiện chính xác lỗ hổng Reentrancy của các phương pháp phát hiện lỗ hổng trong Smart Contract đã tồn tại (Oyente – static analysis – Symbloic execution, ContractWard – sử dụng machine leaerning, …)

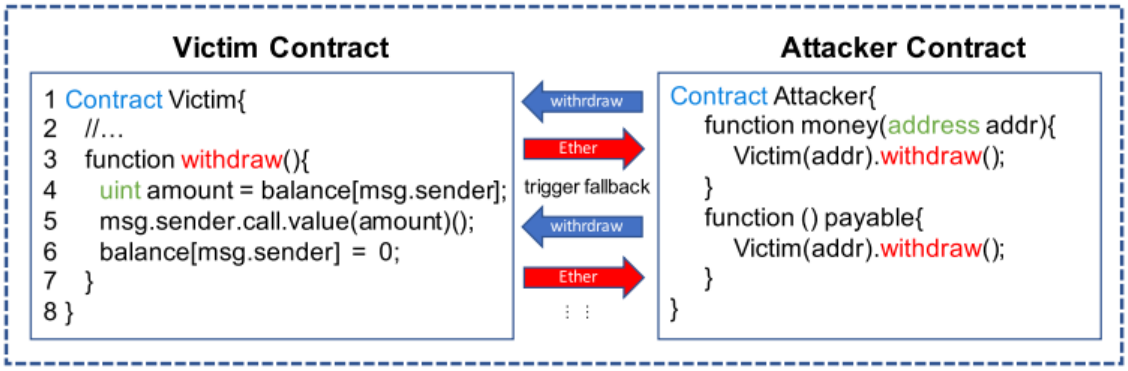

FIGURE 1. An real-world instance of smart contract reentrancy attack.

Hình 1: Reentrancy Vulnerability

Method

Long short-term memory (LSTM) là một kiến trúc artificial recurrent neural network (RNN) được sử dụng trong lĩnh vực Deep learning. Cơ chế hoạt động của LSTM là chỉ ghi nhớ những thông tin liên quan, quan trọng cho việc dự đoán, còn các thông tin khác sẽ được bỏ đi. LSTM hoạt động theo một chiều nhất định (forward direction). Hay nói một cách khác, các mạng này chỉ mang thông tin tính tới thời điểm hiện tại. Tuy nhiên, trong nhiều bài toán NLP thì việc biết thông tin của các timesteps tiếp theo giúp cải thiện rất nhiều kết quả output (Translation, Speech recognition, Handwritten recognition,..). Trong trường hợp này chúng ta có thể sử dụng Bi-directional RNN với việc xử lý thông tin theo cả hai chiều (forward và backward).

Cơ chế Attention là hành động tập trung có chọn lọc vào một vài thứ có liên quan, trong khi đó sẽ bỏ qua những thứ khác có trong mạng nơ-ron sâu (deep neural networks). Cũng có thể hiểu là cơ chế này cho phép mô hình hoá các phần phụ thuộc (dependences) mà không quan tâm đến khoảng cách của chúng trong các chuỗi đầu vào (input sequences) và đầu ra (output sequences). Cơ chế Attention nổi lên như một sự cải tiến so với hệ thống dịch máy dựa trên bộ giải mã (encoder decoder-based neural machine translation) trong xử lý ngôn ngữ tự nhiên (NLP). Nó đã được áp dụng rộng rãi và đạt được sự cải thiện đáng kể trong các nhiệm vụ khác nhau trong xử lý ngôn ngữ tự nhiên như tóm tắt văn bản. Cơ chế Attention có thể hướng sự chú ý đến vị trí chính xác bằng cách sử dụng các dấu hiệu tiềm ẩn trong ngữ cảnh cụ thể (https://arxiv.org/abs/1706.03762). Qua các nghiên cứu gầnđây, cơ chế attention chứng tỏ nó không quá phức tạp nhưng lại mang lại hiệu quả rất cao trong các bài toán dự đoán, đặc biệt là với mô hình Self-attention và sự ra đời của mô hình Tranformer. Điều này cho thấy, việc ứng dụng cơ chế Attention và các mô hình Deep Learning có thể giúp cải thiện khả năng phát hiện chính xác các lỗ hổng bảo mật trong hợp đồng thông minh mà vẫn đảm bảo hiệu suất về thời gian xử lý và tài nguyên tính toán.

⇨ Sử dụng kiến trúc Bidirectional-LSTM và cơ chế attention (Hình 2). Mã nguồn Smart Contract gốc sẽ được vector hóa làm đầu vào cho lớp BLSTM. Sau đó một lớp attention được thêm vào để làm nổi các trọng số quan trọng. Trong quá trình học đặc trưng, cơ chế Attention được sử dụng từ cấp độ word đến cấp độ sentence tương ứng. Phương pháp này tập trung vào việc nắm bắt các word và sentence quan trọng để có được thông tin tối đa về đặc trưng của các Smart Contract. Cuối cùng, nó kết nối các đặc trưng hợp đồng và đặc trưng tài khoản để tạo đại diện đặc trưng cấp tài liệu (document) của Smart Contract và nhận được kết quả phân loại với lớp Softmax.

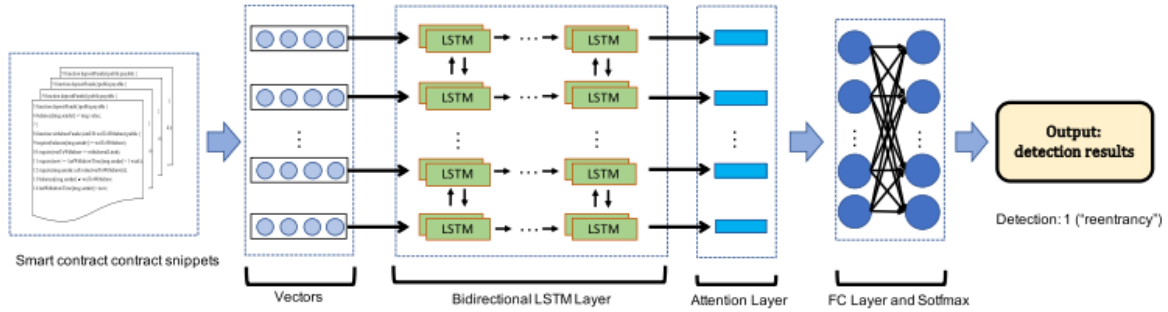

FIGURE 4. **The architecture of our Bidirectional-LSTM with an attention mechanism. First, the vectorized smart *contract snippet* input to the BLSTM layer. Then, an attention layer is added to highlight important weight. Last, through the FC layer and Softmax, the detection result is produced.**

Hình 2: Mô hình phát hiện lỗ hổng Reentrancy dựa trên deep learning và cơ chế attention

Intended experiments:

Sử dụng N-grams và deep learning network để huấn luyện mô hình phát hiện lỗ hổng Reentrancy. Triển khai các mô hình học sâu như: GRU, LSTM (BLSTM+Attention sẽ cố gắng triển khai nếu có thể).

Sử dụng tập dữ liệu được cung cấp trong bài báo (Towards Automated Reentrancy Detection for Smart Contracts Based on Sequential Models) - https://drive.google.com/file/d/1h9aFFSsL7mK4NmVJd4So7IJlFj9u0HRv/view?pli=1 . Tập dữ liệu tập trung vào lỗ hổng Reentrancy trong smart contract (Solidity), chứa 1671 code fragments (197 code gadgets chứa lỗ hổng và 1273 code gadgets không chứa lỗ hổng).

Mô hình sẽ được đánh giá mức độ hiểu quả qua các chỉ số: accuracy (độ chính xác), True Positive Rate (TPR), False Positive Rate (FPR), F1 score.