REENTRANCY SECURITY VULNERABILITY DETECTION IN SMART CONTRACT

UIT
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

1. Động lực

Hợp đồng thông minh (Smart Contract) được viết bằng các ngôn ngữ lập trình không có bảo mật cao như Solidity, … Do đó trong Smart Contract thường tồn tại các lỗ hổng có thể bị attacker khai thác. Trong đó, **REENTRANCY** là lỗ hổng phổ biến (nằm trong danh sách top 10 DASP).

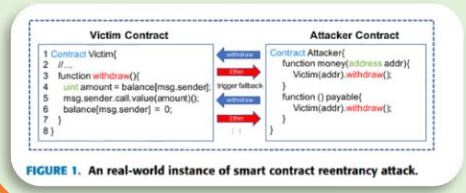

FIGURE 1. An real-world instance of smart contract reentrancy attack.

2. Hướng nghiên cứu

Tự động phát hiện lỗ hổng Reentrancy

- RNN - Deep Learning
- Cơ chế Attention

GROUP 5

1. Lê Hồng Bàng
2. Trần Hoàng Khang
3. Nguyễn Tú Ngọc

3. Phương pháp nghiên cứu

Sử Dụng Kiến Trúc Bidirectional – LSTM & Cơ Chế Attention

3.1. Bidirectional - LSTM

- Cơ chế hoạt động của LSTM là ghi nhớ những thông tin liên quan, quan trọng cho việc dự đoán ; các thông tin khác sẽ được bỏ đi. LSTM hoạt động theo một chiều nhất định (forward direction).
- Tuy nhiên, trong nhiều bài toán NLP thì việc biết thông tin các timesteps tiếp theo giúp cải thiện rất nhiều kết quả output: Translation, Speech recognition, Handwritten recognition,..). Vì thế, **Bidirectional RNN** được sử dụng nhằm xử lý thông tin theo cả hai chiều (forward và backward direction).

3.2. Cơ chế Attention

- Cơ chế hoạt động của Attention là tập trung có chọn lọc vào một vài thứ có liên quan, bỏ qua những thứ không liên quan trong Deep Neural Network.
- Cho phép mô hình hoá các phần phụ thuộc (dependences) mà không quan tâm đến khoảng cách của chúng trong các chuỗi đầu vào và đầu ra (input / output sequences). Hướng đến sự chính xác trong các bài toán dự đoán, đặc biệt là với mô hình **Self-attention** và mô hình **Tranformer**.

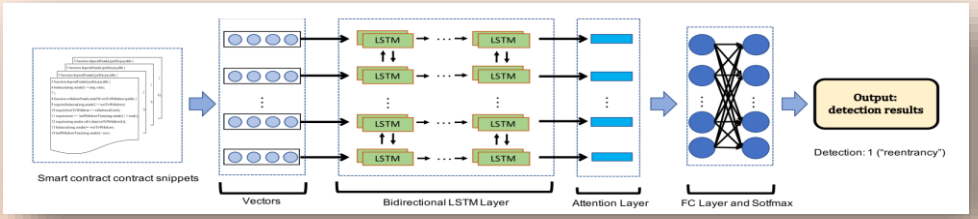

FIGURE 2. **Mô hình phát hiện lỗ hổng Reentrancy dựa trên Deep Learning & cơ chế Attention**

Tóm Lại:

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. 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, 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.

4. Kết quả thực nghiệm

	Accuracy	Precision	FPR	FNR	F1 Score
Simple_RNN	0.777	0.811	0.167	0.279	0.763
LSTM	0.832	0.844	0.15	0.186	0.828
Bi – LSTM	**0.867**	**0.881**	**0.114**	**0.151**	**0.849**
Bi – LSTM - Attention	0.847	0.865	0.128	0.178	0.843

Tập dataset bao gồm:
- 500 Smart Contract không lỗ hổng
- 500 Smart Contract có lỗ hổng

Reentrancy tỉ lệ tập train/test
- 8:2

Learning rate
- 0.002

Dropout
- 0.2

Batch size
- 64

Kết luận:

- Trên tập dataset được thực nghiệm, mô hình Bi - LSTM có kết quả tốt nhất với Accuracy là 86,7%, F1 score là 84,9%.
- Việc dự đoán sai của mô hình Bi - LSTM đối với các Smart Contract có lỗ hổng Reentrancy là 15,1% thấp nhất trong các mô hình trên.

5. Kết luận & Đề xuất

Đồ án hiện chỉ giới hạn trong việc phát hiện ra lỗ hổng **Reentrancy**. Chúng tôi dự định sử dụng các mô hình tuần tự **(Sequential Model)** cho các nghiên cứu sâu hơn về lỗ hổng trong Smart Contract.

Mặt khác, về runtime state, chúng tôi dự định thiết kế một kịch bản tấn công bao quát hơn để tương tác với từng Smart Contract ở các runtime state, nhằm phân tích quá trình thực thi động.