MACHINE LEARNING FOF SECURITY VULNERABILITY DETECTION IN BLOCKCHAIN PROGRAM

LE HONG BANG - 19520396
TRAN HOANG KHANG - 19521671
NGUYEN TU NGOC - 20521665

CCS Concepts: • Computing methodologies → Neural networks.

Additional Key Words and Phrases: blockchain, automated, reentrancy detection, Smart Contracts

ACM Reference Format:
Le Hong Bang - 19520396, Tran Hoang Khang - 19521671, and Nguyen Tu Ngoc - 20521665. 2022. MACHINE LEARNING FOF SECURITY VULNERABILITY DETECTION IN BLOCKCHAIN PROGRAM. 1, 1 (December 2022), 3 pages. https://doi.org/uit.edu.vn

1 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. Bản chất, Smart Contract được viết bằng các ngôn ngữ lập trình có khả năng tồn tại một số điểm hạn chế về bảo mật, điển hình như Solidity,... Do đó trong Smart contract thường tồn tại các lỗ hổng có thể bị attacker khai thác.

Reentrancy là lỗ hổng phổ biến trong Smart Contract (nằm trong danh sách top 10 DASP). 19 tháng 4, 2020, 25 triệu đô đã bị lấy bằng cách khai thác reentrancy trên một nền tảng blockchain "Lendf.me" (một ứng dụng DeFi sử dụng các hợp đồng thông minh để cung cấp dịch vụ cho vay phi tập trung ngay lập tức). Một trường hợp khác, một attacker đã đánh cắp 60 triệu đô la Mỹ bằng cách sử dụng lỗ hổng này trong các tổ chức tự trị phi tập trung (DAO) vào tháng 6, 2016. (3).

Việc 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 nhận diện lỗ hổng trong Smart Contract truyền thống đã tồn tại chẳng hạn như Oyente – static analysis – sử dụng giải pháp Symbolic execution, ContractWard – sử dụng machine learning, ... etc)

Authors' addresses: Le Hong Bang - 19520396, 19520396@gm.uit.edu.vn; Tran Hoang Khang - 19521671, 19521671@gm.uit.edu.vn; Nguyen Tu Ngoc - 20521665, 20521665@gm.uit.edu.vn.

XXXX-XXXX/2022/12-ART $15.00
https://doi.org/uit.edu.vn

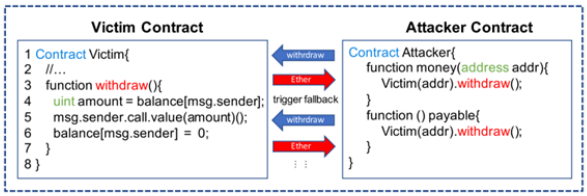

FIGURE 1. An real-world instance of smart contract reentrancy attack.

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

Nói tóm gọn lại, thứ nhất, thiệt hại của lỗ hổng reentrancy trong hợp đồng thông minh có ảnh hưởng mạnh mẽ và không thể phục hồi. Thứ hai, việc phân tích và phát hiện lỗ hổng bảo mật rất khó khăn và đầy thách thức. Cuối cùng, các nghiên cứu trước nay dựa vào phân tích truyền thống sẽ không đạt được mức độ phán đoán chính xác trong khi kết quả sai lệch nghiên về dương tính giả (False Positive) và âm tính giả (False Negative) cao. Do đó, trong bối cảnh hiện tại, các giải pháp áp dụng các vấn đề bảo mật hợp đồng thông minh được yêu cầu khẩn cấp. Đây cũng chính là những luận điểm cốt lõi để chúng tôi thực hiện nghiên cứu và triển khai nhằm đề xuất một hướng giải pháp triệt để và toàn diện hơn.

2 RESEARCH METHODS

2.1 Neuron Model

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 là 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).

Bidirectional LSTM hay BiLSTM là mô hình xử lý trình tự bao gồm hai lớp LSTM. BiLSTM tăng hiệu quả lượng thông tin có sẵn cho mạng, cải thiện ngữ cảnh có sẵn cho thuật toán (ví dụ: biết những từ nào ngay lập tức theo sau và đứng trước một từ trong câu). Hình 2 là một ví dụ cơ bản về mô hình biLSTM với ba bước liên tiếp.

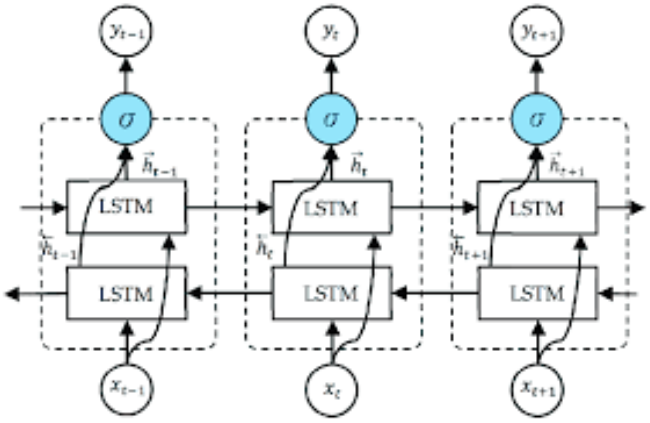

Fig. 2. The architecture of Bidirectional LSTM (BiLSTM)

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 (2). Qua các nghiên cứu gần đây, 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 Smart Contract 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 [Qia+20]. 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, 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.

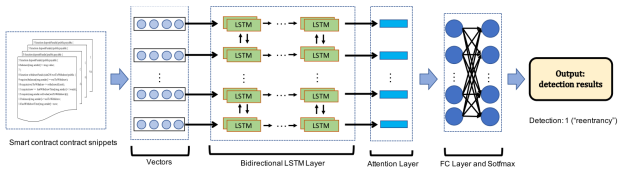

Fig. 3. Mô hình phát hiện lỗ hổng Reentrancy dựa trên deep learning và cơ chế attention

3 EXPERIMENTS

Source code: https://github.com/Bombbom/ML4SecurityProject

3.1 Data source code gathering

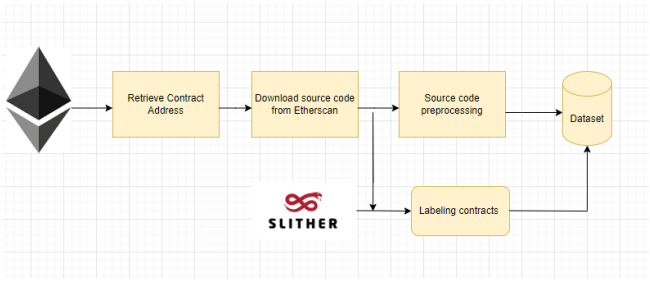

Fig. 4. Qui trình thu thập và gán nhãn smart contract

Dataset được sử dụng để huấn luyện mô hình là các smart contract được thu gom từ trang Etherscan.io và được gán nhãn bằng công cụ Slither (Slither là một công cụ phát hiện các lỗ hổng trong smart contract. Qui trình thu gom và gán nhãn được mô tả theo hình 4.

Tập dataset bao gồm 10000 smart contract đã được xác thực trên Etherscan.io trong đó có 5000 smart contract không chứa bất kỳ lỗ hổng nào và 5000 smart contract có lỗ hổng re-entrancy. Trong quá trình huấn luyện, tập dataset được chia theo tỉ lệ 8:2 cho tập train và tập test.

3.2 Implemented Experiments

- Các tham số cấu hình thực thi (cho tất cả mô hình): Smart contract sau khi được thu thập trên Etherscan.io sẽ

Tham số	Giá Trị
Model Name	"Long-short Term Memory
Learning Nate	0.002
Dropout Rate	0.2
Vector Dimension	300
Epochs	10
Batch Size	64
Threshold	0.5

Bảng 1. Mô tả các tham số cấu hình

được đưa qua quá trình preprocess để làm sạch dữ liệu (xóa các dòng trống, xóa các ký thự non-ASCII và xóa comment). Smart contract sau đó sẽ được trích xuất theo từng dòng và được vector hóa bằng mô hình Word2vec (hình 5). Word2vec là một mô hình Word Embedding nhằm tạo ra các vector đại diện cho mỗi từ sao cho: một từ được biểu diễn bởi một vector có số chiều xác định trước, các từ thuộc cùng một nhóm thì có khoảng các gần nhau trong không gian. Mô hình Word2vec được huấn luyện trên thư viện gensim.

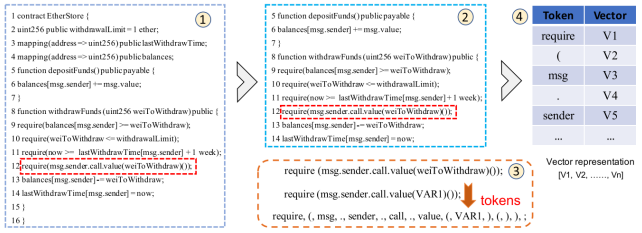

Fig. 5. Vector hóa source code

Sử dụng các mô hình RNN, LSTM, BiLSTM, BiLSTM-attention để huấn luyện mô hình phát hiện lỗ hổng re-entrancy trên kết quả thu được từ mô hình word2vec. Mô hình BiLSTM-attention được trình bày trong hình 3.

3.3 Tiêu chi đánh giá

Mô hình sử dụng deep learning trong việc phát hiện lỗ hổng trên smart contract được đánh giá trên các tiêu chí như: accuracy (ACC(, true positive rate or recall (TPR), false positive rate (FPR) precision (PRE), F1-Score.

4 EVALUATION AND DISCUSSION

4.1 Evaluation

Bảng 2 bao gồm tất cả kết quả của từng mô hình. Các số liệu đầu ra được đo trong 4 lần khác nhau, sau đó tính theo giá trị trung bình.

	Accuracy	Precision	FPR	FNR	F1-score
Standard 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	0.811	0.114	0.151	0.849
Bi-LSTM(Attention)	0.847	0.865	0.128	0.178	0.843

Bảng 2. Kết quả thực nghiệm trung bình trên các mô hình khác nhau

Mô hình được nâng cấp, đổi mới theo thứ tự từ trên xuống dưới. Theo lý thuyết, mô hình Bi-LSTM lẽ ra phải tạo ra kết quả tốt nhất. Bằng cách triển khai thử nghiệm, Bi-LSTM tiêu chuẩn mang lại hiệu suất vượt qua các kiến trúc sư Deep Learning phổ biến cho NLP (Xử lý ngôn ngữ tự nhiên) và Xử lý văn bản và trở thành công cụ tiềm năng và hiệu quả nhất trong bối cảnh này.

4.2 Conclusion & Future Work

Lỗ hổng Reentrancy trong Smart Contract là một loại lỗ hổng có tiềm ẩn cao phụ thuộc vào tư duy và sự hiểu biết của người lập trình. Nếu nhà phát triển tạo ra ứng dụng chưa nắm toàn bộ các case của mã nguồn, rủi ro về lỗ hổng có thể được phát hiện và khai thác. Thiệt hại mà lỗ hổng bảo mật này gây ra là rất lớn và không thể phục hồi hay truy vết. Vì vậy, trong bài báo, chúng tôi đã ứng dụng các mô hình Deep Learning phổ biến để giải quyết, nhận biết và từ đó giảm thiểu rủi ro tối đa khả năng xảy ra của loại tấn công này.

Mô hình được train trên tập dataset với dữ liệu chưa thật sự đa dạng và phong phú. Trong tương lai, chúng tôi sẽ thu thập dữ liệu trên nhiều nguồn sưu tầm khác nhau; đồng thời, xây dựng các mô hình học máy khác và thử nghiệm lại với mục tiêu mang lại hiệu quả cao hơn

5 REFERENCES

(1) Towards Automated Reentrancy Detection for Smart Contracts Based on Sequential Models
(2) https://github.com/Messi-Q/ReChecker
(3) Attention Is All You Need
(4) https://www.coindesk.com/understanding-dao-hack-journalists
(5) https://valid.network/post/the-reentrancy-strikes-again-the-case-of-lendf-me

REFERENCES

[Qia+20] Peng Qian et al. “Towards Automated Reentrancy Detection for Smart Contracts Based on Sequential Models”. In: IEEE Access 8 (2020), pp. 19685–19695. DOI: 10.1109/ACCESS.2020.2969429.

Received 3 October 2022; revised 3 October 2022; accepted 10 October 2022