ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN ĐỨC TẤN

20520751 - ATTT2020

BÁO CÁO THỰC TẬP DOANH NGHIỆP

NGHIÊN CỨU PHÁT HIỆN TẤN CÔNG DAPP TRÊN MẠNG ETHEREUM THÔNG QUA CÁC CHUỖI TRANSACTION

TP. HỒ CHÍ MINH, 12/2023

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO THỰC TẬP DOANH NGHIỆP

NGHIÊN CỨU PHÁT HIỆN TẤN CÔNG DAPP TRÊN MẠNG ETHEREUM THÔNG QUA CÁC CHUỖI TRANSACTION

Công ty thực tập: **Phòng thí nghiệm An toàn thông tin**

Người hướng dẫn tại công ty: **Ngô Khánh Khoa**

Giảng viên hướng dẫn: **ThS. Nguyễn Khánh Thuật**

Tên sinh viên: **Nguyễn Đức Tấn**

MSSV: 20520751 **LỚP: ATTT2020**

TP. HỒ CHÍ MINH, 12/2023

TÊN CƠ QUAN/ĐƠN VỊ/CÔNG TY

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

PHIẾU XÁC NHẬN THỰC TẬP

Công ty:…….xác nhận:

Anh/chị: ..... Sinh ngày: .....

Sinh viên năm thứ…...Khoa: ……Trường: …….....

Đã thực tập tại: ....

Thời gian thực tập: ....

Vị trí thực tập: ....

Nhiệm vụ được giao và hướng dẫn: ....

....

Cán bộ phụ trách và hướng dẫn: ....

Chức vụ: ....

Đánh giá quá trình thực tập của sinh viên:

Các kết quả sinh viên đã thực hiện được: ... .

....

Các tiêu chí đánh giá	**Điểm đánh giá (Theo thang điểm 10)**	**Ghi chú**
1. Năng lực chuyên môn		
2. Chất lượng công việc		
3. Tinh thần trách nhiệm		
4. Tính chủ động sáng tạo		
5. Tính kỷ luật		
6.		
Tổng điểm:		

Đánh giá khác:..... .....

....

…, ngày…tháng …năm …

TRƯỞNG ĐƠN VỊ **NHÂN SỰ** **CÁN BỘ QUẢN LÝ TRỰC TIẾP**

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH
VÀ TRUYỀN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ và tên sinh viên: .....

MSSV: .....

Công ty thực tập: .....

Thời gian thực tập: .....

Vị trí thực tập: .....

Nhiệm vụ được giao: .....

.....

.....

Đánh giá quá trình thực tập của sinh viên:

Các kết quả sinh viên đã thực hiện được: .....

.....

.....

.....

.....

.....

.....

Điểm: ..... Bằng chữ: .....

……., ngày…..tháng …..năm ……

Giáo viên hướng dẫn

MỤC LỤC

MỤC LỤC **1**

DANH MỤC HÌNH VẼ **3**

DANH MỤC BẢNG **5**

DANH MỤC TỪ VIẾT TẮT **6**

LỜI CÁM ƠN **7**

Chương 1. GIỚI THIỆU VỀ CÔNG TY **8**

1.1. Giới thiệu chung 8

1.1.1 Tổng quan về Phòng thí nghiệm An toàn thông tin *8*

1.1.2 Lĩnh vực hoạt động *9*

1.2. Môi trường làm việc 10

Chương 2. GIỚI THIỆU CHƯƠNG TRÌNH THỰC TẬP **11**

2.1. Tổng quan chương trình thực tập 11

2.2. Tóm tắt các kiến thức học được 11

2.3. Thời gian thực tập 11

Chương 3. NỘI DUNG THỰC TẬP **13**

3.1. Các kiến thức nền tảng 13

3.1.1 Blockchain *13*

3.1.2 Solidity *13*

3.1.3 Smart Contract – Dapp *14*

3.1.4 Các dạng tấn công đến Dapp *15*

3.1.5 Pytorch *15*

3.2. Nghiên cứu khoa học 16

3.2.1 Vấn đề được đề cập trong bài báo. *16*

3.2.2 Tổng quan về nội dung.....18

3.2.3 Các bước thực hiện.....18

3.2.4 Phương pháp phân tích và phát hiện tấn công dựa trên các transaction. 18

3.2.5 Mô hình học sâu tự động pháp hiện tấn công dựa trên transaction.....33

Chương 4. KẾT QUẢ CÔNG VIỆC.....35

4.1. Kết quả của mô hình.....35

4.2. Hướng phát triển.....36

Chương 5. TỔNG KẾT, KHÓ KHĂN VÀ HẠN CHẾ 37

5.1. Kỹ năng học được.....37

5.2. Khó khăn.....37

TÀI LIỆU THAM KHẢO 38

DANH MỤC HÌNH VẼ

Hình 1. Logo của Phòng thí nghiệm An toàn thông tin…….7

Hình 2. Hình ảnh văn phòng…….9

Hình 3. Blockchain ……11

Hình 4. Ngôn ngữ Solidity…….12

Hình 5. Cách Smart Contract hoạt động…….12

Hình 6. Pytorch……..14

Hình 7. thông tin hoạt động trong 30 ngày qua của top 5 Dapp có số dư nhiều nhất (tính theo đơn vị dolar) trên dappradar.com……...14

Hình 8. Ghi nhận thiệt hại tài chính do các vụ tấn công được ghi nhận…...……15

Hình 9. Xác định độ dài code của địa chỉ đó để xác định contract…….....19

Hình 10. Xác định self-destructed contract……20

Hình 11. Thu thập transaction của các EOA liên quan…….....20

Hình 12. Thông tin trong transaction tạo contract……..21

Hình 13. Thuật toán Jaccard……...21

Hình 14. Execution trace trên Bloxy.info…….....22

Hình 15. Mô tả về execution trace……..22

Hình 16. Một transaction trong execution trace……..23

Hình 17. Số lượng transaction của mỗi loại tấn công……24

Hình 18. Top 10 sự kiện có số lượng transaction nhiều nhất……...24

Hình 19. Các giai đoạn trong kill chain……..25

Hình 20. Top 3 số lượng method được sử dụng……..27

Hình 21. Top 4 số lượng method được sử dụng……..28

Hình 22. Số lần Dapp bị tấn công…….....28

Hình 23. Số lần Dapp bị tấn công……...29

Hình 24. Top 4 method được sử dụng nhiều nhất…….....29

Hình 25. Số lượng transaction của mỗi loại tấn công……..30

Hình 26. Số lượng transaction tương ứng với mỗi giai đoạn……...31

Hình 27. Sequence transaction cho từng giai đoạn trong một sự kiện cụ thể……31

Hình 28. Tổng quan các bước…….32

Hình 29. Cấu trúc của model……..32

Hình 30. Kết quả demo trong bài báo…….33

Hình 31. Kết quả demo của em……...33

Hình 32. Confusion Matrix.…….....35

Hình 33. Score. ……..35

DANH MỤC BẢNG

Bảng 1. Các báo cáo và phân tích liên quan đến tấn công Dapp……..……18

Bảng 2. Mô tả về các feature trong dataset…….....31

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Viết đầy dủ
Dapp	Decentalized Application
EOA	Externally Owned Accounts
EVM	Ethereum Virtual Machine
tx	Transaction
CTI	Cyber Threat Intelligence

LỜI CÁM ƠN

Trân trọng gửi lời cảm ơn đến Phòng thí nghiệm An Toàn Thông Tin ĐHCNTT – ĐHQGHCM và thầy cô ở phòng E8.1đã tạo điều kiện cho nhóm chúng em có một mùa thực tập thành công.

Hơn thế nữa, nhóm chúng em xin trân thành cảm ơn đến quý thầy cô trường Đại Học Công Nghệ Thông Tin, đặc biệt là quý thầy cô khoa Mạng Máy Tính và Truyền Thông, trong đó chúng em xin gửi lời cảm ơn sâu sắc đến thầy Ngô Khánh Khoa – người hướng dẫn hết sức tận tâm, nhiệt tình và luôn động viên, tạo mọi điều kiện tốt nhất cho chúng em khi thực tập tại Phòng thí nghiệm An Toàn Thông Tin ĐHCNTT – ĐHQGHCM. Những ý kiến đóng góp, hướng dẫn của thầy luôn là nguồn cảm hứng và định hướng cho chúng em ngày càng hoàn thiện, phát triển ứng dụng ngày một tốt hơn và ngày càng hoàn thiện bản thân hơn trong giai đoạn hội nhập và phát triển của đất nước.

Trong quá trình thục hiện nhiệm vụ thực tập, cũng như là trong quá trình làm bài báo cáo thực tập, do kiến thức của chúng em còn hạn chế nên khó có thể tránh khỏi những sai sót không mong muốn, chúng em rất mong quý thầy cô chỉ bảo, đóng góp ý kiến để chúng em ngày càng hoàn thiện hơn về mặt kiến thức. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được ý kiến đóng góp của quý thầy cô, để em học thêm được nhiều kinh nghiệm giúp chúng em trong quá trình tương lai sắp tới.

Cuối cùng, chúng em xin chúc Phòng thí nghiệm An Toàn Thông Tin ngày càng phát triển, đạt được nhiều thành công trong tương lai để có thể tiếp tục dẫn dắt thế hệ trẻ trên con đường An Toàn Thông Tin. Chúng em xin chúc quý thầy cô của trường Đại Học Công Nghệ Thông Tin, đặc biệt là quý thầy cô khoa Mạng Máy Tính và Truyền Thông, và đặc biệt hơn hết đó là thầy Dũng có được nhiều sức khỏe để có thể dốc lòng trong sự nghiệp “trồng người”.

TP HCM, tháng 12 năm 2023

Nguyễn Đức Tấn

Chương 1. GIỚI THIỆU VỀ CÔNG TY

1.1. Giới thiệu chung

1.1.1 Tổng quan về Phòng thí nghiệm An toàn thông tin

Hình 1. Logo của Phòng thí nghiệm An toàn thông tin

Phòng thí nghiệm An Toàn Thông Tin (InSec Lab) thuộc Trường Đại học Công nghệ Thông tin – ĐHQG TP.HCM được thành lập theo quyết định số 19/QĐ-ĐHCNTT-TCHC ngày 19/01/2016, hướng đến việc xây dựng một phòng thí nghiệm chuyên sâu về an toàn thông tin, có chức năng nghiên cứu khoa học trong lĩnh vực an toàn thông tin, giải quyết các nhu cầu đặt ra bởi thực tiễn cũng như các giải pháp cho tương lai, tham gia đào tạo đại học, sau đại học, hợp tác quốc tế, chuyển giao công nghệ trong lĩnh vực an toàn thông tin.

Phòng Thí nghiệm An toàn thông tin (UIT InSec Lab) là một môi trường nghiên cứu, thực nghiệm chuyên dụng dành cho các nghiên cứu về an toàn thông tin, hỗ trợ sinh viên, nghiên cứu viên tham gia vào các hoạt động liên quan đến an toàn thông tin trên các hệ thống mạng với các tài nguyên máy tính, môi trường thực nghiệm, cũng như các dịch vụ ứng dụng được kiểm soát, mà không ảnh hưởng đến các môi trường mạng khác.

Cơ sở hạ tầng phục vụ nghiên cứu của Phòng Thí nghiệm bao gồm hệ thống server mạnh mẽ với nhiều cơ chế phân phối, xử lý dữ liệu đa dạng và các dịch vụ về an toàn thông tin.

UIT InSec Lab hướng đến việc cung cấp một nền tảng hỗ trợ và thúc đẩy sự hợp tác giữa các nghiên cứu viên trong môi trường học thuật, chính phủ điện tử và doanh nghiệp. Hệ thống phần mềm quản lý cơ sở hạ tầng của Phòng Thí nghiệm An toàn thông tin được đặt lại Phòng E8.1 (tòa nhà E) và Data Center (tòa nhà A) - UIT.

1.1.2 Lĩnh vực hoạt động

1.1.2.1 Định hướng hoạt động chính

- Thực hiện các nhiệm vụ nghiên cứu khoa học liên quan đến an toàn thông tin:
    - Triển khai các đề án nghiên cứu khoa học và công nghệ có tính cấp thiết, tính đi trước trên cơ sở bám sát định hướng phát triển, mục tiêu kinh tế, xã hội Quốc gia và tiến bộ khoa học hiện đại trên thế giới.
    - Tập hợp và phát triển đội ngũ cán bộ khoa học công nghệ có trình độ cao, ưu tiên các nghiên cứu, hợp tác quốc tế, tạo điều kiện thuận lợi triển khai các nghiên cứu theo định hướng về An toàn thông tin
- Thực hiện tư vấn, nghiên cứu giải pháp và chuyển giao công nghệ nhằm giải quyết vấn đề kĩ thuật, công nghệ cụ thể có liên quan đến an toàn thông tin do địa phương, doanh nghiệp đặt hàng.

1.1.2.2 Những hướng nghiên cứu chính tại Phòng thí nghiệm

Phòng thí nghiệm An toàn thông tin tập trung vào các chủ đề nghiên cứu chính như sau:

- SPS: Software-defined programmable security (SDN, NFV, Cloud, Edge).
- Lập trình an toàn: Giải pháp kiểm thử xâm nhập, bảo mật thông tin và tính riêng tư cho người dùng trong các ứng dụng (end-to-end encryption, pentesting, software vulnerablity,....)
- Điều tra bằng chứng số, tội phạm số (digital forensics).
- AI security and AI-based security: An toàn thông tin, tính riêng tư dữ liệu cho các mô hình AI & Ứng dụng trí tuệ nhân tạo (deep learning,

Generative Adversarial Networks) cho bài toán an ninh trên không gian mạng.

- Internet malware/botnet/APT detection, defense, and analysis.
- Blockchain (các công nghệ nền tảng, giao thức và các ứng dụng thực tế)
- Mobile and IoT security

1.2. Môi trường làm việc

Môi trường làm việc đày tính sáng tạo, học thuật, là môi trường tốt để các bạn thực tập sinh cũng như là sinh viên có thể phát triển tốt trong lĩnh vực An Toàn Thông Tin.

Hình 2. Hình ảnh văn phòng

Chương 2. GIỚI THIỆU CHƯƠNG TRÌNH THỰC TẬP

2.1. Tổng quan chương trình thực tập

Tìm hiểu và nghiên cứu blockchain, cụ thể là tìm hiểu các loại tấn công đến các Dapp từ các transaction. Tìm hiểu và triển khai mô hình học sâu để tự phát hiện các loại tấn công từ các transaction.

2.2. Tóm tắt các kiến thức học được

Khi tham gia thực tập, em đã có được một số kiến thức sau

- Kiến thức cơ bản về blockchain và cách nó hoạt động.
- Các loại tấn công phổ biến trên blockchain, đặc biệt là liên quan đến các Dapp (smart contract).
- Khả năng phân tích, trích lọc các thông tin và xử lý data.
- Hiểu biết cơ bản về ngôn ngữ lập trình Solidity.
- Hiểu biết về mô hình học sâu, cách sử dụng thư viện Pytorch để lập trình mô hình học sâu.

2.3. Thời gian thực tập

- Thời gian thực tập: từ ngày 02/10/2023 đến ngày 31/12/2022.
- Hình thức làm việc: online và offline.
- Yêu cầu thực hiện việc offline tại văn phòng 4 buổi/tuần.

2.4. Nhiệm vụ được giao

Em được giao nhiệm vụ tìm hiểu lý thuyết cơ bản của Blockchain, thực hành các bài lab để hiểu rõ về cơ chế hoạt động cũng như một số điểm yếu của chúng. Sau đó nghiên cứu bài báo khoa học về giải pháp phân tích tấn công dựa trên các transaction (sẽ được đề cập rõ trong phần sau).

- Tuần 1: Làm quen với môi trường làm việc, hỗ trợ nhóm thực tập trước đó về công việc debug.

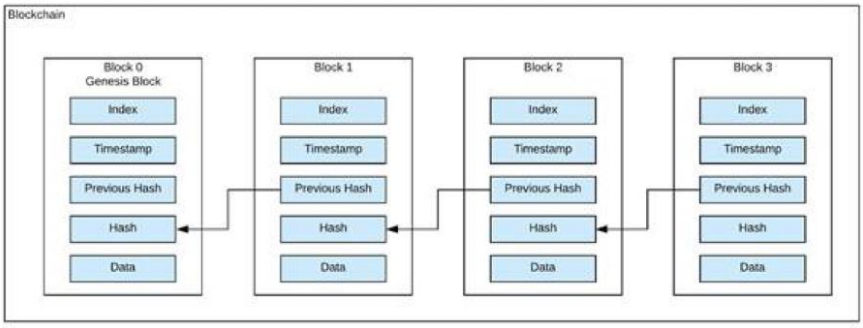
- Tuần 2: Học lý thuyết và làm lab cơ bản về khái niệm blockchain, NFT, smart contract.
- Tuần 3: Nghiên cứu bài báo được giao "Evil Under the Sun Understanding and Discovering Attacks on Ethereum"
- Tuần 4: Học lý thuyết về các tấn công trên blockchain và các chế phòng thủ.
- Tuần 5: Nghiên cứu bài báo và thu thập dữ liệu gồm các report ghi nhận tấn công đến các Dapp.
- Tuần 6: Phân tích và trích xuất các transaction liên quan đến tấn công Dapp từ các report để tạo dataset.
- Tuần 7: Thu thập thêm dữ liệu để mở rộng dataset.
- Tuần 8: Tìm hiểu các công cụ khác giúp thu thập dataset và dựng đồ thị mô tả hành vi tấn công.
- Tuần 9: Tìm hiểu các paper liên quan đến phát hiện lỗ hổng trong Smart contract, thực hiện lại phương pháp được đề cập trong paper.
- Tuần 10: Viết code cho mô hình học sâu phát hiện tấn công dựa trên các transaction.
- Tuần 11: Hoàn thiện code và kiểm tra kết quả mô hình.
- Tuần 12: Tiếp tục hoàn thiện và cải tiến mô hình.

Chương 3. **NỘI DUNG THỰC TẬP**

3.1. Các kiến thức nền tảng

3.1.1 Blockchain

Blockchain là một công nghệ lưu trữ phân tán, sử dụng các khối thông tin (block) được liên kết với nhau để tạo thành một chuỗi. Đặc điểm nổi bật của blockchain là tính toàn vẹn cao, vì dữ liệu được lưu trữ trong blockchain không thể bị thay đổi hoặc phá hủy, và chỉ có thể thêm các khối chứa dữ liệu mới khi tất cả các node trong hệ thống đồng ý. Bên cạnh đó, blockchain còn có tính sẵn sàng cao, ngay cả khi một phần của hệ thống gặp sự cố, các node khác vẫn có thể hoạt động bình thường. Điều này giúp đảm bảo tính tin cậy của thông tin trong blockchain. Một số nền tảng public blockchain nổi tiếng như Ethereum, IBM,…



Hình 3. Blockchain.

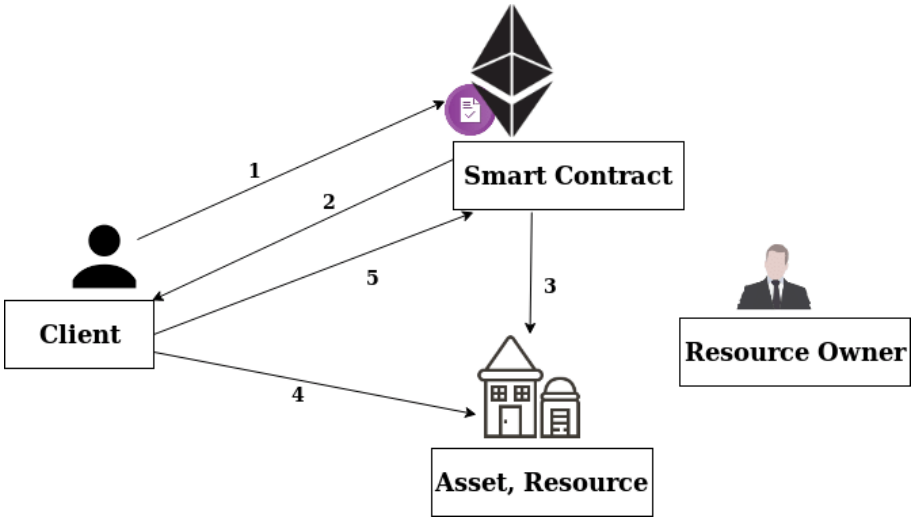
3.1.2 Solidity

Solidity là một ngôn ngữ lập trình hướng đối tượng được sử dụng để xây dựng các ứng dụng trên nền tảng Ethereum và EVM chain. Ngôn ngữ này được phát triển bởi đội ngũ Ethereum Network và được sử dụng để tạo các hợp đồng thông minh. Solidity hỗ trợ việc tạo hợp đồng thông minh và biên dịch mã máy cấp thấp trên Ethereum Virtual Machine (EVM). Nó cũng có nhiều đặc điểm tương tự với ngôn ngữ lập trình C và C++, giúp người dùng dễ dàng học và hiểu. Solidity cung cấp các khái niệm cơ bản như biến, hàm, lớp, phép toán số học và xử lý chuỗi, giúp người phát triển xây dựng các ứng dụng dựa trên blockchain.

Hình 4. Ngôn ngữ Solidity.

3.1.3 Smart Contract – Dapp

Smart contract (hợp đồng thông minh) trên nền tảng Ethereum là một chương trình tự thực thi được lưu trữ trên blockchain Ethereum. Nó được viết bằng ngôn ngữ lập trình Solidity và được triển khai trên Ethereum Virtual Machine (EVM). Thường thấy ở việc tạo ra các Dapp – là các ứng dụng phi tập trung (Decentralize App), để phục vụ các mục đích khác nhau như các trò chơi, nền tảng tài chính (DeFi), thị trường phi tập trung (DEX),… Điều khác biệt so với các app thông thường là không có sự can thiệp từ bên thứ ba, khi triển khai lên EVM thì không thể sửa đổi, và mọi hoạt động của app điều minh bạch và có thể hoàn toàn theo dõi được thông qua các giao dịch trên Ethereum.



Hình 5. Cách Smart Contract hoạt động.

3.1.4 Các dạng tấn công đến Dapp

Trong môi trường thực tế, bất kỳ hệ thống và công nghệ nào cũng phải tồn tại lỗ hổng trong quá trình hoạt động, dẫn đến các lỗ hổng khác nhau để cho kẻ tấn công khai thác. Điều đó đúng với cả Dapp, đa phần mục tiêu cuối cùng của các dạng tấn công này thường là cố đánh cắp tài chính mà các Dapp đó đang nắm giữ, hoặc làm cản trở quá trình hoạt động của nó. Một số loại tấn công phổ biến như:

- DoS: Attacker tạo ra nhiều giao dịch đến các Dapp nhằm mục đích khiến Dapp sử dụng hết `gas`, từ đó Dapp không thể thực hiện bất kỳ giao dịch nào khác.
- Bad randomness: Attacker dự đoán được giá trị ngẫu nhiên do Dapp tạo ra (do cơ chế pseudo-random yếu), mục đích là nhận được các phần thưởng của việc dự đoán đúng.
- Integer overflow and underflow: Attacker truyền cho Dapp các input nằm ngoài phạm vi tối đa hoặc tối thiểu mà kiểu dữ liệu đó có thể lưu trữ, dẫn đến tràn dữ liệu để thực hiện các hành vi khác.
- Reentrancy attack: Contract có thể gọi public function của các contract khác, điều khiển hoặc lợi dụng chúng để thực hiện các hành vi khác.
- Improper authentication: Xác thực không đúng cách, attacker lợi dụng lỗ hổng trong quá trình xác thực của Dapp để truy cập đến các tài nguyên hoặc điều khiển Dapp.

3.1.5 Pytorch

PyTorch là một framework học máy dựa trên thư viện Torch, được sử dụng cho các ứng dụng như thị giác máy tính và xử lý ngôn ngữ tự nhiên, được phát triển ban đầu bởi Meta AI và hiện nay là một phần của Linux Foundation. Nó là phần mềm mã nguồn mở và miễn phí được phát hành dưới giấy phép BSD được sửa đổi.

Hình 6. Pytorch.

3.2. Nghiên cứu khoa học

Em thực hiện nghiên cứu dựa trên cơ sở chính từ bài báo:

Su, L., Shen, X., Du, X., Liao, X., Wang, X., Xing, L., & Liu, B. (2021). Evil under the sun: understanding and discovering attacks on Ethereum decentralized applications. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 1307-1324).

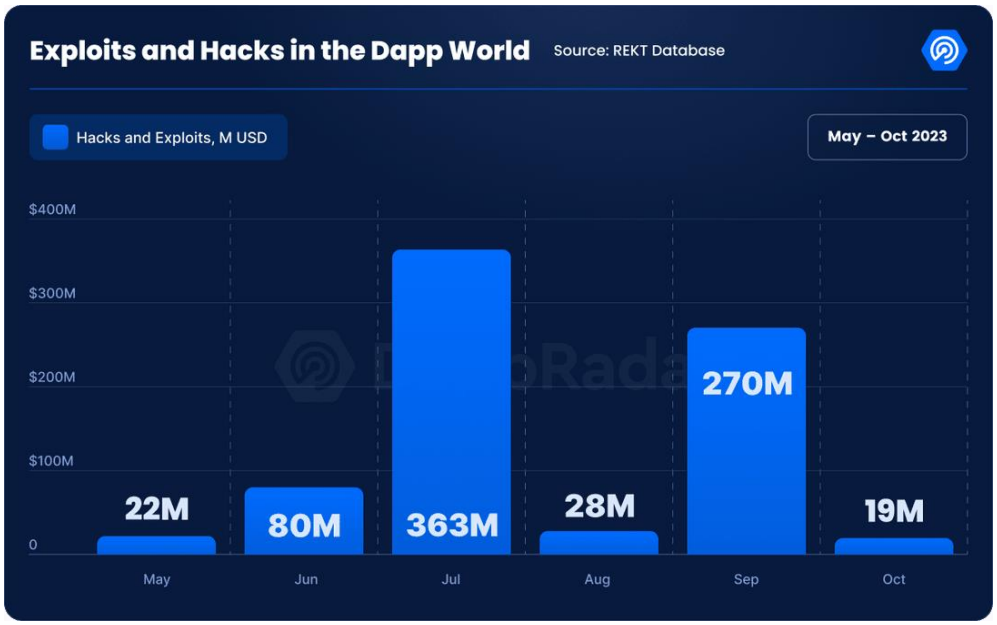
3.2.1 Vấn đề được đề cập trong bài báo.

Việc áp dụng công nghệ Blockchain đã trở nên phổ biến, nhiều loại hình công nghệ và kỹ thuật được thực hiện dựa trên nó. Đối với người dùng, công nghệ quen thuộc đối với họ có lẽ là Dapp – các App chạy trên mạng Ethereum, nó cung cấp nhiều loại dịch vụ khác nhau như game, tài chính, lưu trữ,… Có khoảng 3.000 Dapp chạy trên mạng Ethereum, phục vụ cho khoảng 63.000 user (thu thập từ năm 2019).

#	Name	Balance	UAW	% UAW	Volume	% Volume
Ad	MetaWin Ethereum	$1.17M	3.42k	-10.31%	$4.3M	-34.04%
1	ETH2 Deposit Contract Ethereum	$57.6B	3.72k	-3.54%	$758.85M	+21.26%
2	Minswap Cardano	$4.22B	19.93k	+2.61%	$1.4B	+5.67%
3	Polygon POS Bridge Ethereum	$3.19B	8.59k	+37.36%	$237.36M	+11.09%
4	Midas Miner BNB Chain	$1.98B	8.55k	+26.76%	$561.8k	+172.5%

Hình 7. thông tin hoạt động trong 30 ngày qua của top 5 Dapp có số dư nhiều nhất (tính theo đơn vị dolar) trên dappradar.com.

Có khoảng 3000 Dapp đang hoạt động (tháng 10/2023) và phục vụ hàng chục nghìn user sử dụng các Dapp mỗi này, nên có hàng triệu giao dịch được tạo ra với tổng giá trị lên đến hàng triệu đô . Vì vậy, chúng là những "miếng mồi ngon" cho những kẻ tấn công khai thác để trục lợi. Theo ghi nhận trong năm 2023 , đã ghi nhận 22 các báo cáo sự cố khác nhau liên quan đến tấn công với tổng thiệt hại về mặt tài chính lên đến hàng triệu dollar.



Hình 8. Ghi nhận thiệt hại tài chính do các vụ tấn công được ghi nhận (nguồn: dappradar.com – 11/2023)

Tuy đã có rất nhiều báo cáo giải thích về các vụ tấn công, tuy nhiên hầu như nó chỉ mang tính tường minh là chính, họ chỉ nêu điểm yếu được khai thác là gì, một số báo cáo thì có thực hiện lại tấn công, cuối cùng là nêu hậu quả (thiệt hại tài chính) do các loại tấn công đó để lại. Rất ít báo cáo đề cập đến việc thu thập các *cyber threat intelligence (CTI)* để phân tích các hành vi, chiến lược mà kẻ tấn công đã thực hiện, và thường không đưa ra các khuyến nghị để ngăn chặn hoặc làm giảm khả năng bị tấn công trong tương lai.

3.2.2 Tổng quan về nội dung

Với các vấn đề trên được đề cập, bài báo đã thực hiện nghiên cứu và phân tích các vụ tấn công đã được ghi nhận, trích xuất các thông tin khác ngoài báo cáo, từ đó mang lại nhiều nhận định khác nhau về chiến lược mà các kẻ tấn công đã thực hiện.

Tất cả hoạt động tương tác giữa các chủ thể trên mạng Etherum đều được ghi nhận và lưu lại, được gọi là transaction, nên khi Dapp bị tấn công, các transaction này sẽ là các thông tin vô cùng chất lượng để phục vụ cho quá trình điều tra và truy vết. Trên cơ sở đó, nhóm tác giả đã tìm kiếm các transaction liên quan đến vụ tấn công cụ thể, sau đó phân tích sự tương tác giữa chúng và Dapp, từ đó dự đoán hành vi và chiến lược mà kẻ tấn công đã sử dụng. Sau khi phân tích tất cả, họ đưa ra *kill chain*, gồm các giai đoạn cơ bản mà kẻ tấn công sẽ thực hiện khi tấn công một Dapp nào đó. Cuối cùng họ đề xuất một mô hình học sâu nhằm tự động hóa quá trình giám sát và phát hiện các hành vi tấn công xảy ra trên các transaction.

3.2.3 Các bước thực hiện

Theo như bài báo đề cập, gồm có hai bước chính:

- Thu thập dataset gồm tất cả các transaction liên quan đến các báo cáo đã ghi nhận, sau đó thực hiện các bước xử lý để gom nhóm các transaction, phân loại chúng dựa trên loại tấn công. Sau đó thực hiện phân tích các hành vi tương tác giữa kẻ tấn công và Dapp, từ đó xác định từng transaction đó đang ở giai đoạn nào trong *kill chain*.
- Đề xuất và phát triển mô hình học sâu giúp tự động phân loại và phát hiện tấn trong trên các transaction đến Dapp.

3.2.4 Phương pháp phân tích và phát hiện tấn công dựa trên các transaction.

Phương pháp này gồm 3 nhiệm vụ chính:

- Thu thập transaction từ các báo cáo tấn công được ghi nhận.

- Mở rộng thông tin bằng cách tìm các transaction khác liên quan đến kẻ tấn công và Dapp trước và sau thời điểm ghi nhận tấn công. Và tìm kiếm các EOA, smart contract khác liên quan hoặc có sự tương đồng về hành vi hoặc đặc điểm (ví dụ như một số smart contract khác tương tự được tạo ra để khai thác cùng một điểm yếu nhưng trên các Dapp khác mà không được đề cập trong báo cáo)
- Phân tích các transaction này để tìm hiểu ý nghĩa của từng tương tác giữa chúng, từ đó xác định các đặc trưng cụ thể, từ những đặc trưng đó, ta có thể phân chia các giai đoạn được thực hiện để tấn công

A. Nhiệm vụ 1: Thu thập thông tin

Nguồn thu thập dữ liệu từ các security report, blog và tin tức. Các loại dữ liệu cụ thể được thu thập gồm:

- Tất cả transction hash được đề cập trong nguồn trên.
- Attacker EOAs.
- Địa chỉ của Dapp, exploit contract.

Theo như bài báo, họ tìm xác định được có 42 Dapp bị tấn công, 20 exploit contract và 48 attacker EOA, từ năm 2016 đến năm 2018.

Còn đối với em, trong quá trình tìm kiếm, em đã trích xuất các thông tin này từ 20 báo cáo khác nhau, xác định được 22 Dapp, 36 attacker EOA cùng với tất cả transaction hash được đề cập trong đó. Dưới đây là các báo cáo được phân tích:

Ngày báo cáo	Tiêu đề báo cáo	Link
23/04/2018	New batchOverflow Bug in Multiple ERC20 Smart Contracts (CVE-2018–10299)	https://peckshield.medium.com/alert-new-batchoverflow-bug-in-multiple-erc20-smart-contracts-cve-2018-10299-511067db6536
25/04/2018	Integer Overflow (i.e., proxyOverflow Bug) Found in Multiple ERC20 Smart Contracts (CVE-2018–10376)	https://peckshield.medium.com/integer-overflow-i-e-proxyoverflow-bug-found-in-multiple-erc20-smart-contracts-14fecfba2759
28/04/2018	Your Tokens Are Mine: A Suspicious Scam Token in A Top Exchange	https://peckshield.medium.com/your-tokens-are-mine-a-suspicious-scam-token-in-a-top-exchange-5e864075f7e9
10/05/2018	New multiOverflow Bug Identified in Multiple ERC20 Smart Contracts (CVE-2018–10706)	https://peckshield.medium.com/new-multioverflow-bug-identified-in-multiple-erc20-smart-contracts-cve-2018-10706-8e55946c252c
08/07/2018	An Inspection on AMMBR(AMR) Bug	https://medium.com/coinmonks/an-inspection-on-ammbr-amr-bug-a53b4050d52
12/07/2018	Multiple Ethereum contracts have high-risk vulnerabilities in unlimited issuance	https://www.jianshu.com/p/a1237c5cfebb
20/06/2017	Parity Multisig Wallet Hack	https://medium.com/aeternity-com/parity-multisig-wallet-hack-47cc507d964d

18/05/2018	New burnOverflow Bug Identified in Multiple ERC20 Smart Contracts (CVE-2018–11239)	https://peckshield.medium.com/new-burnoverflow-bug-identified-in-multiple-erc20-smart-contracts-cve-2018-11239-52cc4f821694
24/06/2018	New ceoAnyone Bug Identified in Multiple Crypto Game Smart Contracts (CVE-2018–11329)	https://medium.com/@peckshield/new-ceoanyone-bug-identified-in-multiple-crypto-game-smart-contracts-cve-2018-11329-898cdceac7e0
24/05/2018	New allowAnyone Bug Identified in Multiple ERC20 Smart Contracts (CVE-2018–11397, CVE-2018–11398)	https://peckshield.medium.com/new-allowanyone-bug-identified-in-multiple-erc20-smart-contracts-20d935b5e7ff
24/06/2018	New evilReflex Bug Identified in Multiple ERC20 Smart Contracts (CVE-2018–12702, CVE-2018–12703)	https://peckshield.medium.com/new-evilreflex-bug-identified-in-multiple-erc20-smart-contracts-63ebece2c94f
12/04/2016	Hijack Rubixi	https://bitcointalk.org/index.php?topic=1400536.60
04/01/2017	Vulnerability in StandardToken.sol's implementation of transferFrom()	https://github.com/ether-camp/virtual-accelerator/issues/8
13/02/2018	Tricked by a honeypot contract or beaten by another hacker. What happened?	https://www.reddit.com/r/ethdev/comments/7x5rwr/tricked_by_a_honeypot_contract_or_beaten_by/
12/06/2018	The Bitcoin Exchange Coinrail was stolen, and the loss of more than 40 million Bitcoin fell below $ 7,000	https://t.cj.sina.com.cn/articles/view/6429008901/17f32e40500100719d
18/08/2018	An analysis of advanced attack technology for FOMO3D gaming airdrops	https://paper.seebug.org/672/
25/07/2018	Pwning Fomo3D Revealed: Iterative, Pre-Calculated Contract Creation For Airdrop Prizes!	https://peckshield.medium.com/pwning-fomo3d-revealed-iterative-pre-calculated-contract-creation-for-airdrop-prizes-31944a01387e
21/03/2018	Revealing the "smuggling" vulnerability that has been dormant for many years in Ethereum, global hackers are frantically stealing coins	https://paper.seebug.org/547/
24/07/2018	Gibraltar blockchain exchange RKT has an overflow vulnerability	https://www.freebuf.com/vuls/178496.html
12/09/2018	[DappPub] FairDice Version 2 is back! An open-sourced, rig-resistant and verifiably fair EOS dice game !	https://www.reddit.com/r/eos/comments/9f5o6a/dapppubfairdice_version_2_is_back_an_opensourced/
18/06/2016	Analysis of the DAO exploit	https://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

Bảng 1. Các báo cáo và phân tích liên quan đến tấn công Dapp.

Tất cả chúng liên quan đến 5 dạng tấn công là bad randomness, DoS, integer overflow/underflow, reentrancy, improper authentication.

B. Nhiệm vụ 2: Mở rộng thông tin

Nhiệm vụ này nhằm mục đích tái hiện lại cách tấn công thông qua các transaction, việc này, theo nhóm tác giả nói là muốn hình dung được toàn bộ ngữ cảnh, từ đó tìm hiểu được ý nghĩa và giúp trích xuất các thông tin hữu ích cho nhiệm vụ thứ 3. Để thực hiện được diều này, nhóm tác giả tìm các EOA khác liên quan đến các báo cáo, đó là các EOA được tạo mới, được gọi hoặc có chuyển tiền đến các attacker EOA và exploit contract. Các dữ liệu đó được thu thập từ trường *from/to* trong transaction. Khi thu thập phải phân biệt đâu là EOA và đâu là exploit contract. Địa chỉ EOA là địa chỉ user nên sẽ không chứa code, tuy nhiên nếu exploit contract "tự hủy" (self-destructed contract) thì cũng còn chứa code bên trong. Sau đó là thu thập các contract liên quan đến vụ tấn công đó, đó là các contract được attacker EOA đã gọi trong

khoảng 3 ngày (trước và sau timestamp của exploit transaction). Sau đó, sử dụng thuật toán Jaccard để tính độ tương đồng của các contract này với contract được đề cập trong báo cáo, khi độ tương đồng là 90% thì hai contract đó tương đồng với nhau. Việc so sánh này nhằm đảm rằng các contract đó cũng dùng cho mục đích tấn công tương tự chứ không phải cho mục đích khác. Vấn đề ở đây là đối với self-destructed contract, vì nó đã tự hủy nên sẽ mất hết code bên trong, để giải quyết vấn đề này, tác giả đề xuất giải pháp là khôi phục runtime code từ transaction tạo contract. Với cách đó, nhóm tác giả đã xác định được 58,555 exploit transaction, 45 exploit contract, 227 attacker EOA và 56 Dapp.

Phía trên là mô tả các bước làm, còn bây giờ là cách mà em đã thực hiện chúng. Khi đọc phân tích các báo cáo (bảng 1), để mở rộng dữ liệu, em cần làm 2 việc:

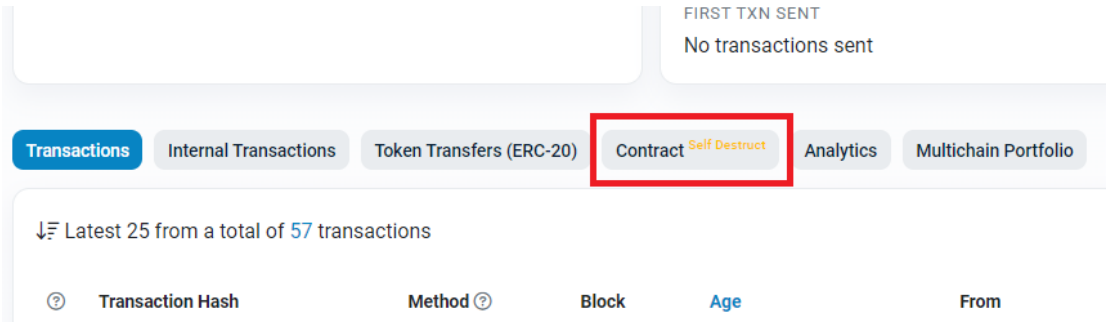
- Xác định các địa chỉ của các các EOA (ngoài attacker EOA) tương tác với exploit contract.
- Xác định các contract khác tương đồng với exploit contract.

Nhưng đầu tiên, em cần làm rõ việc là làm thế nào để phân biệt đâu là địa chỉ của EOA, đâu là địa chỉ của smart contract và self-destructed contract. Như đã đề cập thì EOA sẽ không chứa code bên trong, nên nếu chứa code thì nó sẽ là smart contract. Dưới đây là code xác định:

```
# Get the bytecode of the address
code = w3.eth.get_code(Web3.to_checksum_address(address))
lenght_code = len(code)

if lenght_code > 1:
    return "contract"
```

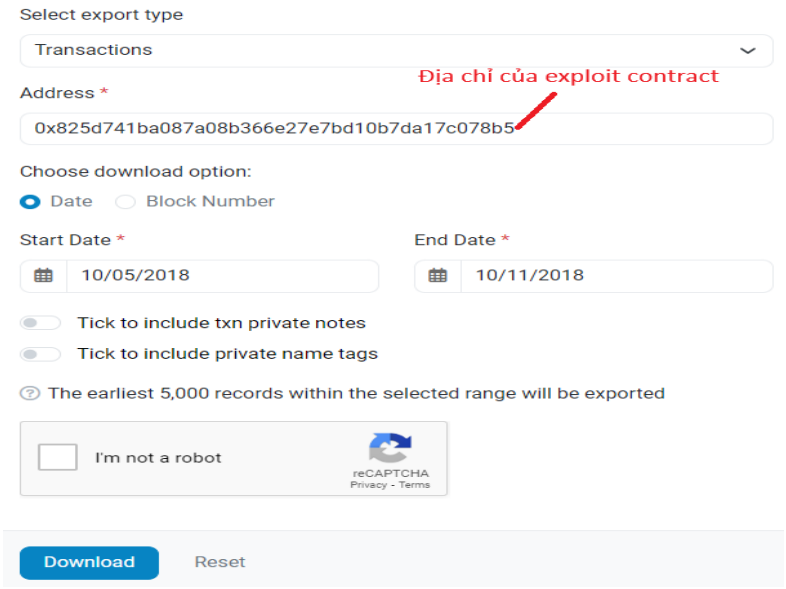
Hình 9. Xác định độ dài code của địa chỉ đó để xác định contract.

Nhưng đối với self-destructed contract thì `**lenght_code**` (hình 9) sẽ là 0, nên em sẽ tìm kiếm địa chỉ đó trên Etherscan.io, nếu xuất hiện mục "*contract*" thì nó là self-destructed contract, ngược lại thì sẽ là EOA.



Hình 10. Xác định self-destructed contract.

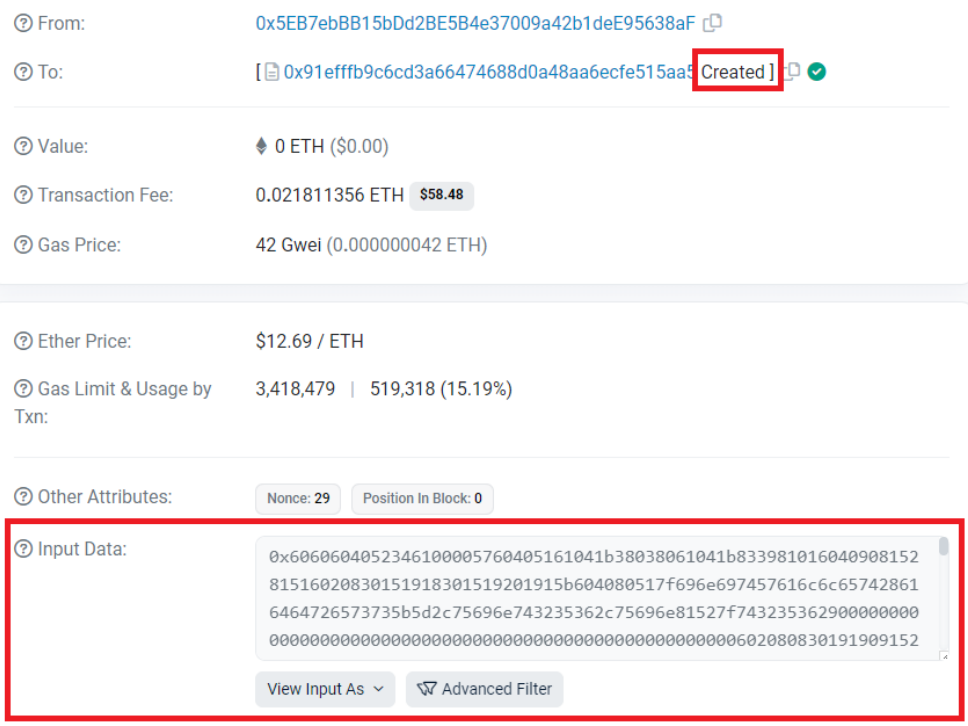
Tiếp theo là trích xuất toàn bộ EOA tương tác với exploit contract. Em tiếp tục sử dụng Etherscan.io, để download toàn bộ transaction của exploit contract (trước và sau thời diểm ghi nhận tấn công 3 ngày). Em sẽ xác định timestamp của exploit transaction và địa chỉ của attacker EOA, sau đó em dùng Etherscan.io để download toàn bộ transaction của attacker EOA trong khoảng thời gian trước và sau timestamp đó khoảng 3 ngày (tổng cộng là 6 ngày). Sau đó, em duyệt toàn bộ transaction vừa download được, trích xuất địa chỉ của tất cả EOA (trong trường *from/to* của transaction) đã tương tác với exploit contract.



Hình 11. Thu thập transaction của các EOA liên quan.

Cuối cùng là thu thập các contract khác mà attacker EOA đã tương tác. Áp dụng cách tương tự như ở hình 11 nhưng thay vào đó là địa chỉ của attacker EOA, em sẽ trích xuất toàn bộ contract trong khoảng thời gian trước và sau exploit transaction

khoảng 1 tuần. Sau đó, em sẽ trích xuất code của contract đó, còn đối với self-destructed contract, em sẽ khôi phục runtime code (là đoạn hex nằm trong trường *input data*) của transaction đã tạo ra contract đó.



Hình 12. Thông tin trong transaction tạo contract.

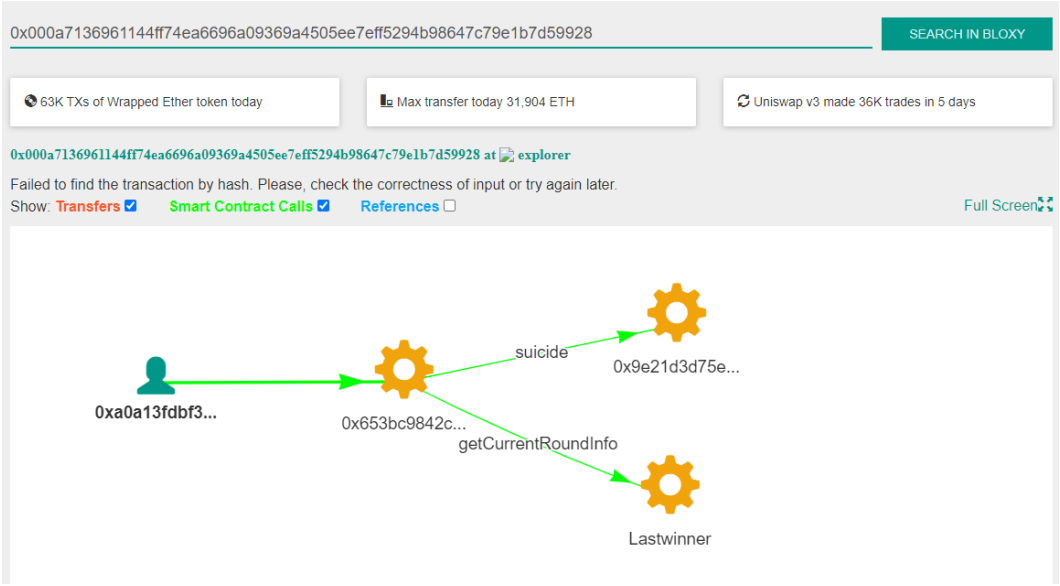
Sau đó, em sử dụng thư việc Octopus để tạo execution graph của các contract trên và dùng thuật toán Jaccard để tìm tự tương đồng. Dưới đây là code mẫu thực thi:

```
exploit_contract_hexcode = "0x606060405263ffffffff60e060020a600035041663202023d48
contract_hexcode = "0x6080604052348015610010576000 80fd5b503360008061010000a8154817

def jaccard_similarity(exploit_contract_hexcode, contract_hexcode):
    def create_execute_graph(hexcode):
        cfg = EthereumCFG(hexcode)
        return CFGGraph(cfg)

    exploit_contract_graph = create_execute_graph(exploit_contract_hexcode)
    contract_graph = create_execute_graph(contract_hexcode)

    edges1 = set(exploit_contract_graph.edges)
    edges2 = set(contract_graph.edges)

    intersection = edges1.intersection(edges2)
    union = edges1.union(edges2)
    return len(intersection) / len(union)
```

Hình 13. Thuật toán Jaccard.

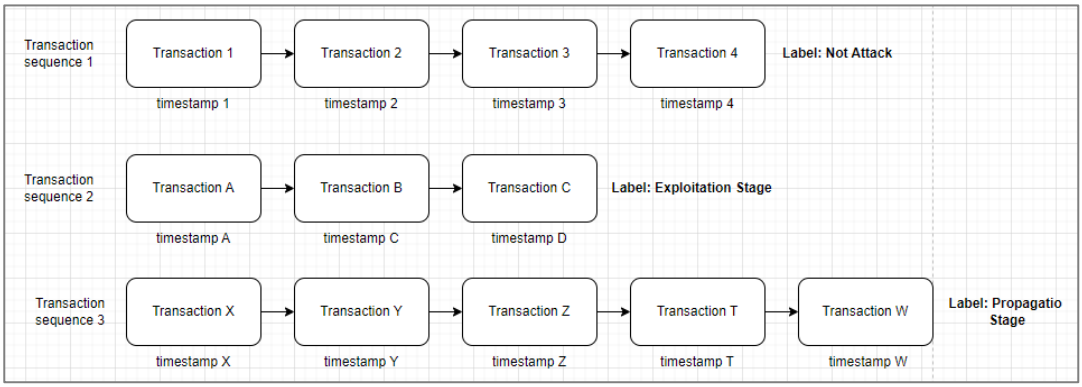
Liên tục lặp lại các phương pháp trên, em mở rộng dataset của mình từ khoảng dưới 100 transaction lên thành 29394 transaction (là tổng số transaction của toàn bộ báo cáo ở bảng 1), 113 exploit contract, 94 attacker EOA và 79 Dapp.

Sau đó, với mỗi transaction, em sẽ tìm execution trace của nó, dữ liệu này được em tìm trên Bloxy.info.



Hình 14. Execution trace trên Bloxy.info.

Về cơ bản, execution trace là một tập các transaction khác nhau (sequence transaction theo timestamp), được phân tích là có liên quan đến một sự kiện nào đó.



Hình 15. Mô tả về execution trace.

Các transaction này sẽ được sử dụng để dựng *trace graph* để biểu diễn mối quan hệ gữa các EOA và contract (hình 13), *trace graph* là một đồ thị có hướng, trong đó

mỗi đỉnh trong đồ thị (node) là EOA hoặc contract, các cạnh có hướng (directed edge) sẽ là tên method được sử dụng trong transaction đó. Quay lại với hình 13, biểu tượng con người có trong hình là một EOA và các biểu tượng hình bánh răng là contract hoặc Dapp. Đồ thị đang mô tả tấn công của attacker EOA (0x0a13...) đến Dapp ***Lastwinner***, đầu tiên attacker sẽ tạo contract 0x653b…, sau đó gọi method *suicide* để hủy contract 0x9e21…, lúc này balance của contract 0x9e21… sẽ được chuyển về cho contract 0x653b… Cuối cùng contract 0x653b… gọi đến phương thức *getCurrentRoundInfo* của Dapp ***Lastwinner*** để khai thác tấn công. Lưu ý rằng đây chỉ là một phần trong toàn bộ quy trình tấn công, bởi vì vậy nên mới phải thu thập nhiều transaction khác thì mới có thể tái hiện được toàn bộ tấn công.

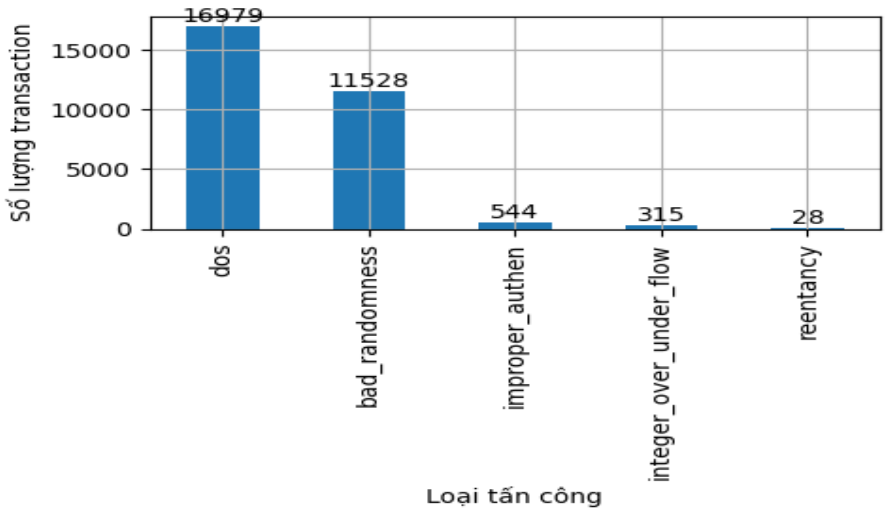
Về mặt dữ liệu, một execution trace được lấy trên Bloxy.info sẽ có dạng JSON. Từ dạng dữ liệu này, em sử dụng thư viện *networkx* để dựng thành trace graph.


```
[
    {
        "id": "0x653bc9842cb54fa5454645be2eed7265f9fb9523",
        "label": "0x653bc9842c...",
        "group": "smart_contract",
        "title": "Smart Contract 0x653bc9842cb54fa5454645be2eed7265f9fb9523",
        "link": "/address/0x653bc9842cb54fa5454645be2eed7265f9fb9523"
    },
    {
        "id": "0xdd9fd6b6f8f7ea932997992bbe67eabb3e316f3c",
        "label": "Lastwinner",
        "group": "smart_contract",
        "title": "Smart Contract 0xdd9fd6b6f8f7ea932997992bbe67eabb3e316f3c",
        "link": "/address/0xdd9fd6b6f8f7ea932997992bbe67eabb3e316f3c"
    },
    "call",
    "0x653bc9842cb54fa5454645be2eed7265f9fb9523",
    "0xdd9fd6b6f8f7ea932997992bbe67eabb3e316f3c",
    "getCurrentRoundInfo",
    0.0,
    1
],
```


Hình 16. Một transaction trong execution trace.

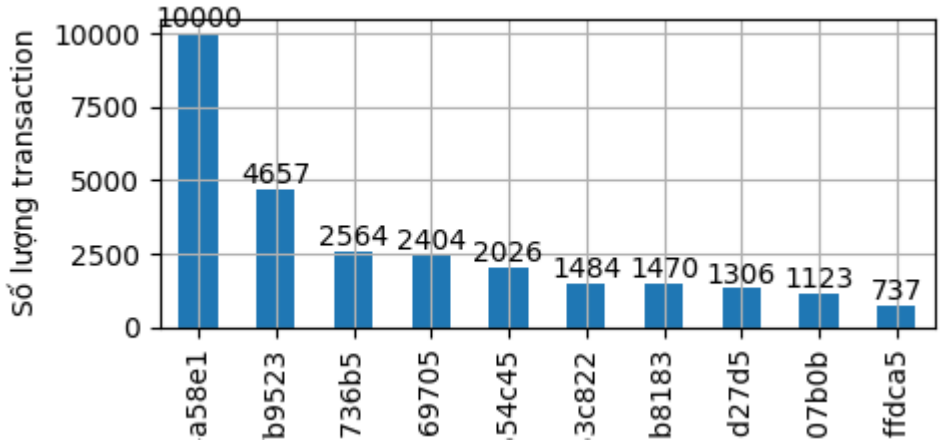
C. Nhiệm vụ 3: Phân tích và tạo dataset

Từ hai nhiệm vụ trên, em đã tìm được tổng số 29394 transaction, phân loại thành 5 loại tấn công khác nhau.



Hình 17. Số lượng transaction của mỗi loại tấn công.

Số lượng transaction của mỗi báo cáo được ghi nhận. Dưới đây là bảng thông kê top 10 tự kiện có số lượng transacrtion nhiều nhất, trong đó trục hoành là 5 hex cuối của mỗi attacker EOA thuộc sự kiện đó.



Hình 18. Top 10 sự kiện có số lượng transaction nhiều nhất.

Tiếp theo, em sẽ xác định xem từng transaction sẽ thuộc giai đoạn nào trong *kill chain*. Theo định nghĩa của nhóm tác giả, *kill chain* gồm tập hợp các giai đoạn cơ bản khác nhau được attacker EOA thực hiện khi tấn công một Dapp nào đó, theo phân tích thì có 4 giai đoạn chính trong *kill chain*:

Hình 19. Các giai đoạn trong kill chain.

- Attack preparation (giai đoạn 1): Đây là giai đoạn đầu tiên, attacker EOA sẽ thực hiện các bước kiểm tra để đảm bảo lỗ hổng trên Dapp có thể khai thác được. Tạo exploit contract và chuẩn bị số dư để thực hiện tấn công.
- Exploitation (giai đoạn 2): Giai đoạn tấn công, sau khi đảm bảo có thể khai thác lỗ hổng thành công, attacker EOA sẽ thực thiện tấn công toàn diện lên Dapp đó.
- Propagation (giai đoạn 3): Giai đoạn lan truyền, attacker EOA sẽ thực hiện khai thác lỗ hổng tương tự trên các Dapp khác.
- Completion (giai đoạn 4): Khi khai thác thành công, attacker sẽ thực hiện rút số dư từ các Dapp đó, sau đó hủy các exploit contract để xóa dấu vết.

Theo như tìm hiểu của em thì cách xác định từng giai đoạn sẽ dựa vào 2 yếu tố chính:

- Thời gian các transaction được tạo ra: Em sử dụng thời điểm ghi nhận tấn công từ các báo cáo để làm cột mốc chính. Các transaction trước cột mốc này sẽ thường thuộc giai đoạn 1, các transaction được báo cáo ghi nhận sẽ thuộc giai đoạn 2, các transaction tương tác với các Dapp khác sau vụ tấn công được ghi nhận sẽ là giai đoạn 3 và các transaction thực hiện rút tiền, hủy contract sẽ là giai đoạn 4.
- Tên method được sử dụng trong các transaction: Đó là các method phổ biến được sử dụng trong từng giai đoạn khác nhau, nhờ việc nhận diện loại method được sử dụng, em có thể xác định chính xác hơn

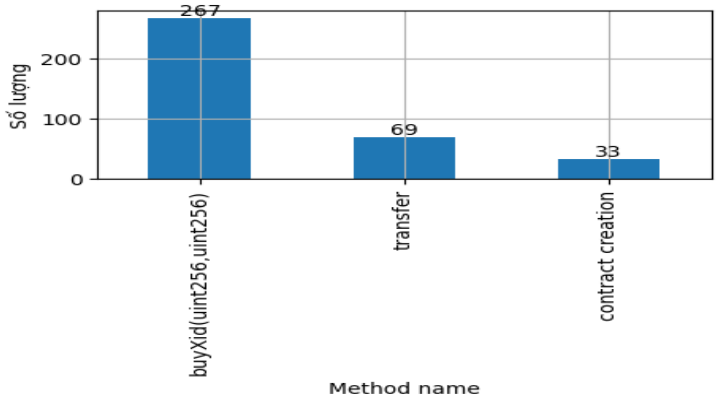
Một số vấn đề em gặp phải trong suốt quá trình phân tích:

- Ranh giới xác định giữa các giai đoạn: Vì không có bất kỳ báo cáo nào nói rõ về chúng, nên hướng giải quyết của em là dựa vào tính đặc trưng của từng giai đoạn trong bài báo đúc kết ra để xác định.
- Tấn công được ghi nhận trong các báo cáo thực tế không thuộc giai đoạn 2 (tấn công), mà là ở giai đoạn 3 (lan truyền): Vì về cơ bản thì các kỹ thuật được sử dụng để tấn công ở hai giai đoạn này khá giống nhau, nó chỉ khác nhau về mục tiêu tấn công. Để giải quyết vấn đề này, em sẽ truy vết toàn bộ transaction của attacker EOA trong khoảng 1 tuần trước thời điểm được báo cáo, nếu xuất hiện các transaction sử dụng kỹ thuật nhưng với Dapp khác sớm nhất, thì đó sẽ thời điểm thuộc giai đoạn 2.
- Một số vấn đề về thu thập dữ liệu: Một số trang web có các chính sách, qui định nhằm giới hạn hoặc không cho phép lấy một thông tin khác liên quan đến Dapp và transaction.

Dưới đây là một số đặc trưng của mỗi giai đoạn mà em tìm hiểu được:

i. Giai đoạn 1. Attack preparation:

Như đã đề cập, đây là giai đoạn đầu tiên, được dùng để kiểm tra khả năng khai thác lỗ hổng, giai đoạn này và giai đoạn 2 có thể giống nhau về kỷ thuật khai thác nên rất dễ bị nhầm lẫn chỗ này. Vì vậy, thay vào đó xem sẽ xác định địa chỉ đích của transaction đó, vì nếu attacker kiểm tra ngay trên Dapp và rút được tiền, thì đó sẽ là một tấn công chứ không còn là kiểm thử nữa. Nên dặc trưng trong giai đoạn này là có rất nhiều exploit contract hoặc các contract có chức năng tương tự Dapp được dựng lên để thực hiện kiểm tra lỗ hổng và cải tiến các exploit contract để giúp cho giai đoạn 2 hoạt động trơn tru nhất có thể. Ngoài ra giai đoạn này còn được dùng để chuẩn bị số dư (gas, cryptocurrency,…) để phục vụ cho các giai đoạn tấn công phía sau. Dưới đây là top 3 các method được sử dụng chủ yếu trong giai đoạn này:

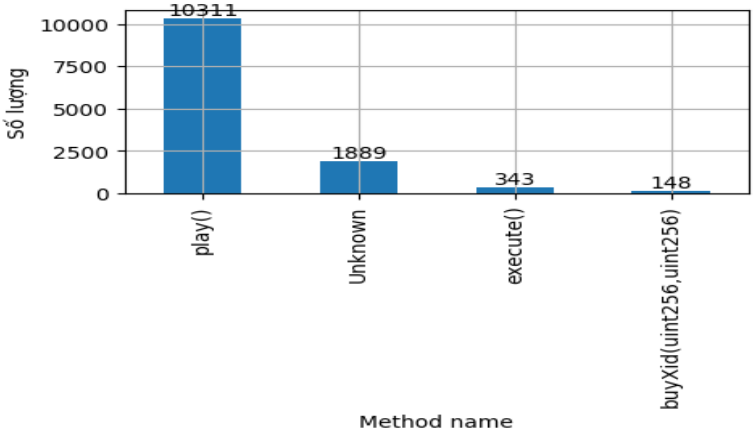


Hình 20. Top 3 số lượng method được sử dụng.

Theo như hình, method ***buyXid(unint256, uint256)*** được sử dụng nhiều nhất, bởi vì method này chứa lỗ hổng bad randomness, có rất nhiều báo cáo về dạng tấn công này tại thời điểm năm 2017 (phổ nhất là Dapp ***Fomo3D***) cùng với sự xuất hiện các PoC hướng dẫn nên nó trở nên phổ biến. Tiếp theo là method ***transfer*** được dùng để chuyển tiền nhằm mục đích chuẩn bị số dư cho việc kiểm thử và tấn công ở các giai đoạn sau. Cuối cùng là method tạo contract, mục đích là dựng các contract để kiểm thử, tạo và cải tiến các exploit contract. Rất ít

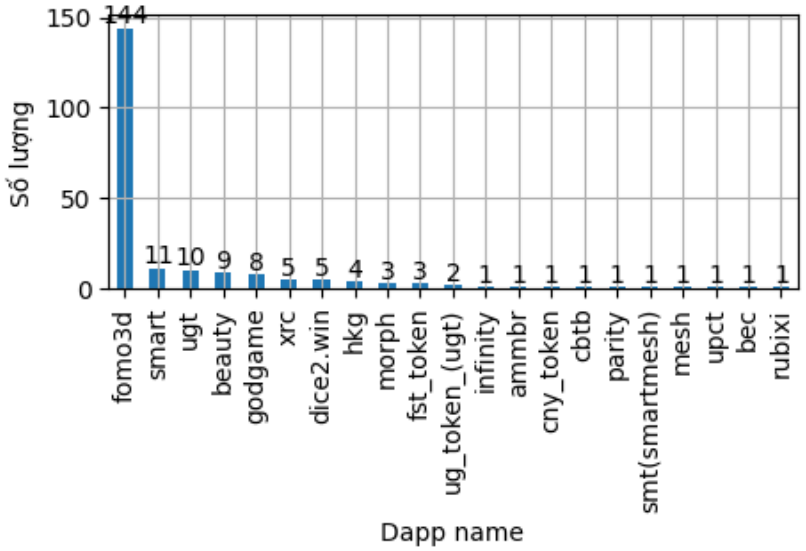
ii. Giai đoạn 2. Tấn công:

Ở giai đoạn này, các transaction tấn công sẽ nhắm thẳng vào các Dapp, bao gồm tất cả các thông tin đã được ghi nhận trong các báo cáo. Dưới đây là top 4 các method được sử dụng chủ yếu trong giai đoạn này:



Hình 21. Top 4 số lượng method được sử dụng.

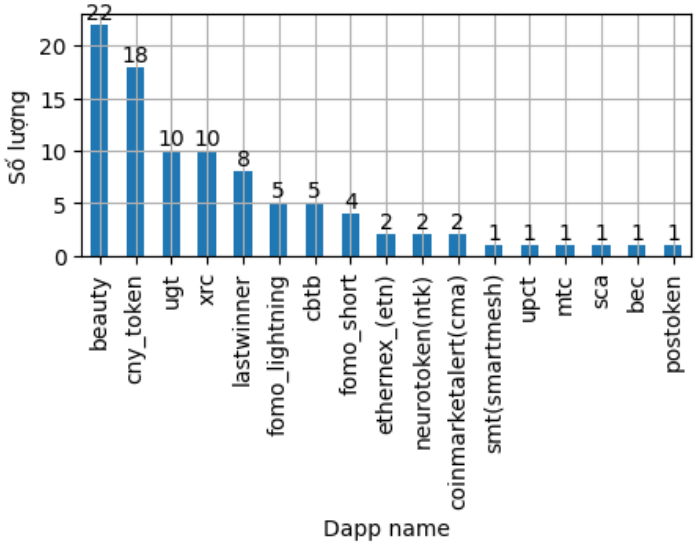
Method tên ***play()*** chứa lỗ hổng cho phép thực hiện tấn công DoS lên Dapp, vì tính chất của loại tấn công này nên cần phải tạo ra nhiều transaction để gọi nó. Số lần mà các Dapp bị tấn công trong giai đoạn này là:



Hình 22. Số lần Dapp bị tấn công.

iii. Giai đoạn 3. Lan truyền:

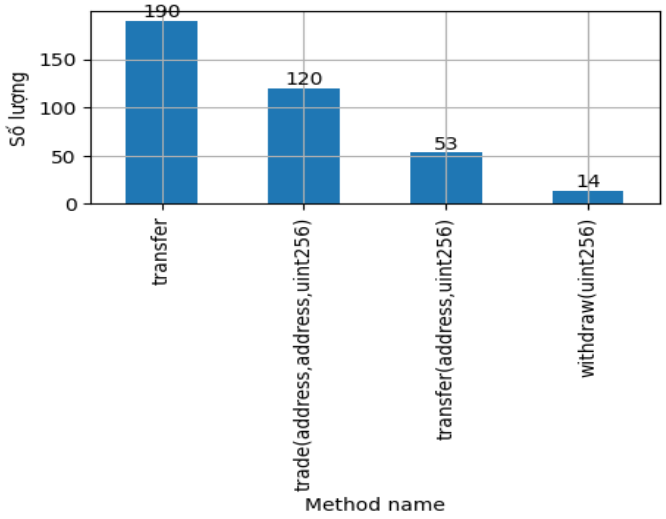
Áp dụng các kỹ thuật tấn công tương tự ở giai đoạn 2 để tấn công các Dapp khác có cùng lỗ hổng. Dưới đây là số lần các Dapp khác bị tấn công.



Hình 23. Số lần Dapp bị tấn công.

iv. Giai đoạn 4. Hoàn thành tấn công:

Đây là giai đoạn attacker hưởng "thành quả" bằng cách chiếm đoạt số dư của các Dapp về ví của chúng, thực hiện các nhiều giao dịch truyển tiền khác nhau nhằm mục đích gây rối, làm mất dấu cho quá trình điều tra, và thực hiện hủy các exploit contract để xóa dấu vết. Các method được sử dụng chủ yếu ở giai đoạn này:



Hình 24. Top 4 method được sử dụng nhiều nhất.

Như hình 23, cả 4 method này đều liên quan đến việc vận chuyển số dư từ địa chỉ này sang địa chỉ khác.

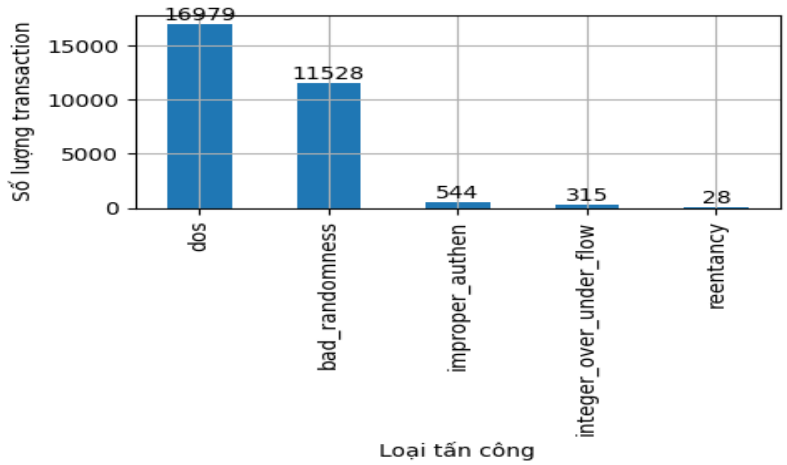
Dưới đây là tóm tắt về dataset của em:

Tên cột	Miêu tả
Seed_event	Địa chỉ của EOA, các sample có cùng giá trị này thì sẽ liên quan đến cùng 1 sự kiện
Timestamp	Thời điểm transaction được tạo
Tx_hash	Hash của transaction
Tx_status	Trạng thái của transaction (0 là thành công, 1 là thất bại)
From_addr	Địa chỉ gửi của transaction
From_addr_type	Loại địa chỉ: EOA, contract, exchange
To_addr	Địa chỉ nhận của transaction
To_addr_type	Loại địa chỉ: EOA, contract, exchange
Method_name	Tên method được sử dụng trong transaction
Method_hex	Hex code của method đó
Value	Số dư được chuyển đi

Input	Thông tin khác như tham số cho hàm,….
Gas	Giá trị yêu cầu để thực hiện giao dịch
Gas_price	Số ETH (Gwei) phải trả cho số gas trên
Nonce	Số nonce được tạo dựa trên transaction đó
Stage	Giai đoạn tấn công trong *kill chain*
Attack_type	Tên loại tấn công

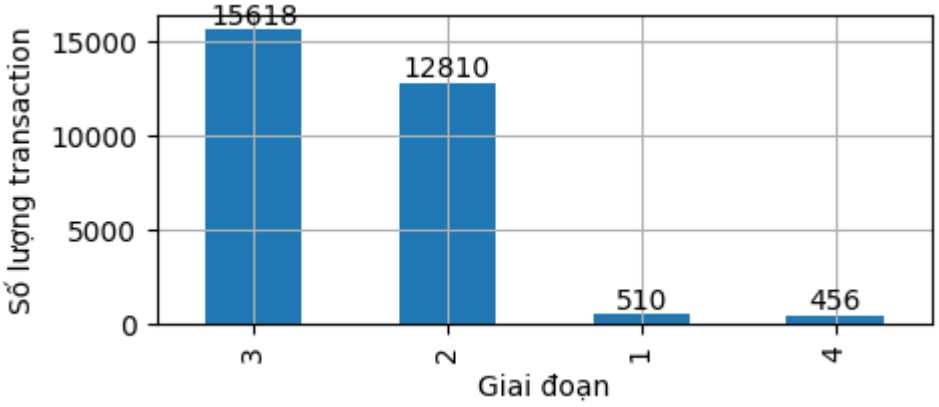
Bảng 2. Mô tả về các feature trong dataset.

Số lượng transaction của mỗi loại tấn công:



Hình 25. Số lượng transaction của mỗi loại tấn công.

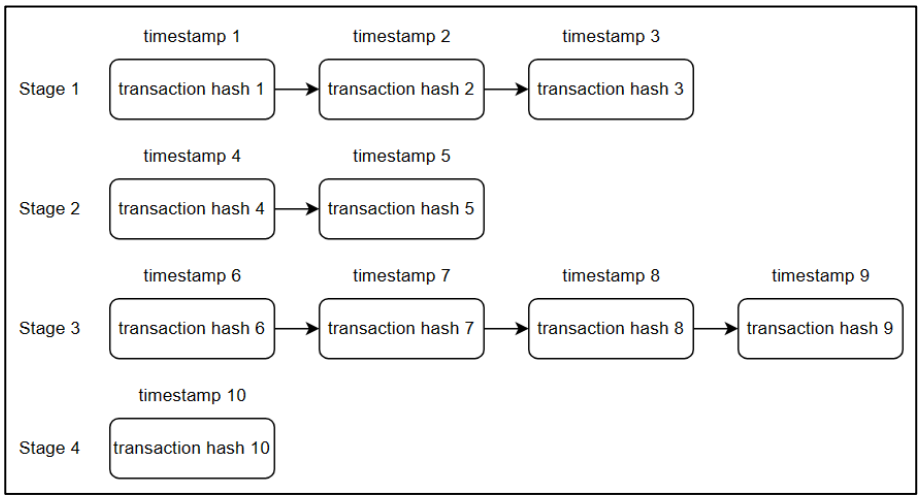
Số lượng transaction cho mỗi giai đoạn:



Hình 26. Số lượng transaction tương ứng với mỗi giai đoạn.

Cuối cùng là xây dựng sequence dataset. Như đã biết thì mỗi sự kiện tấn công đều có đủ 4 giai đoạn trong *kill chain*. Vì vậy với mỗi sự kiện, em sẽ gom nhóm tất cả

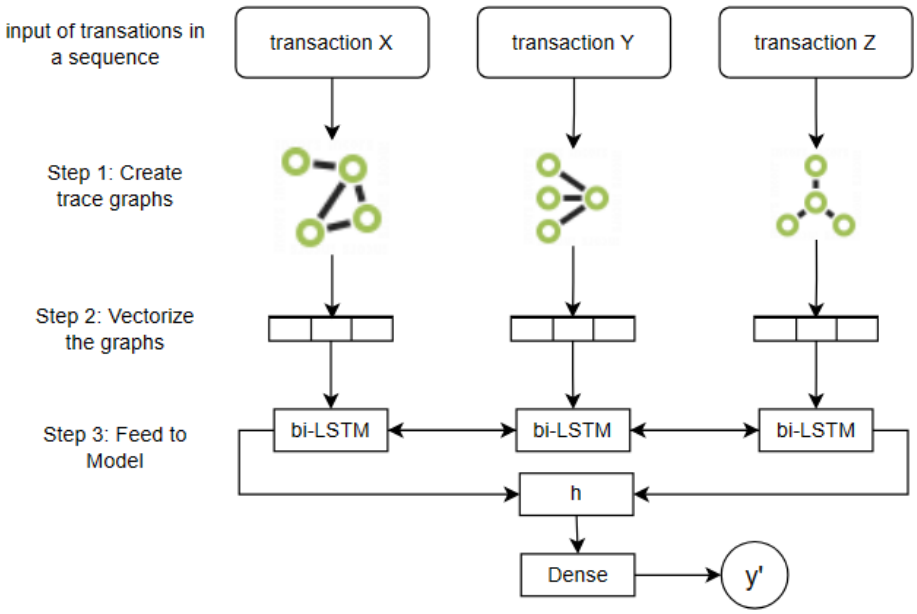
transaction cùng chung lại với nhau, sắp xếp theo timestamp. Từ đó, em có thể tạo được sequence transaction cho mỗi giai đoạn trong từng sự kiện khác nhau.



Hình 27. Sequence transaction cho từng giai đoạn trong một sự kiện cụ thể.

3.2.5 Mô hình học sâu tự động pháp hiện tấn công dựa trên transaction

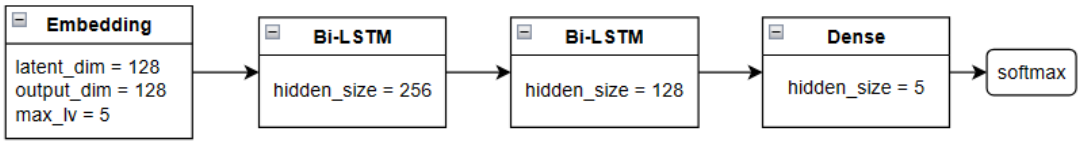
Model theo bài báo sử dụng được sử dụng là sequence classification, vì vậy nó mới phù hợp sequence dataset của em. Dataset sẽ được xử lý và đưa vào model bi-LSTM để train. Dưới đây là mô hình hóa các bước thực hiện:



Hình 28. Tổng quan các bước.

- Bước 1: Với mỗi transaction trong sequence, em sẽ lấy data mô tả các trace dưới dạng Json, sau đó dùng thư viện **networkx** để tạo trace graph dựa trên số data đó (đã giải thích ở hình 14).
- Bước 2: Em sẽ chuyển graph này thành các vector trước đưa và model để train. Ở đây em sử thư viện **Node2Vec** để thực hiện công việc này.
- Bước 3: Cuối cùng là đưa các vector này vào mô hình bi-LSTM để train và predict. Outptut của mô hình cho sequence transaction đưa vào đang ở giai đoạn nào của kill chain.

Cấu trúc của model:



Hình 29. Cấu trúc của model.

Các tham số khác:

- Learning rate = 0.001
- Loss là MSE
- Batch size: 128

Chương 4. KẾT QUẢ CÔNG VIỆC

4.1. Kết quả của mô hình

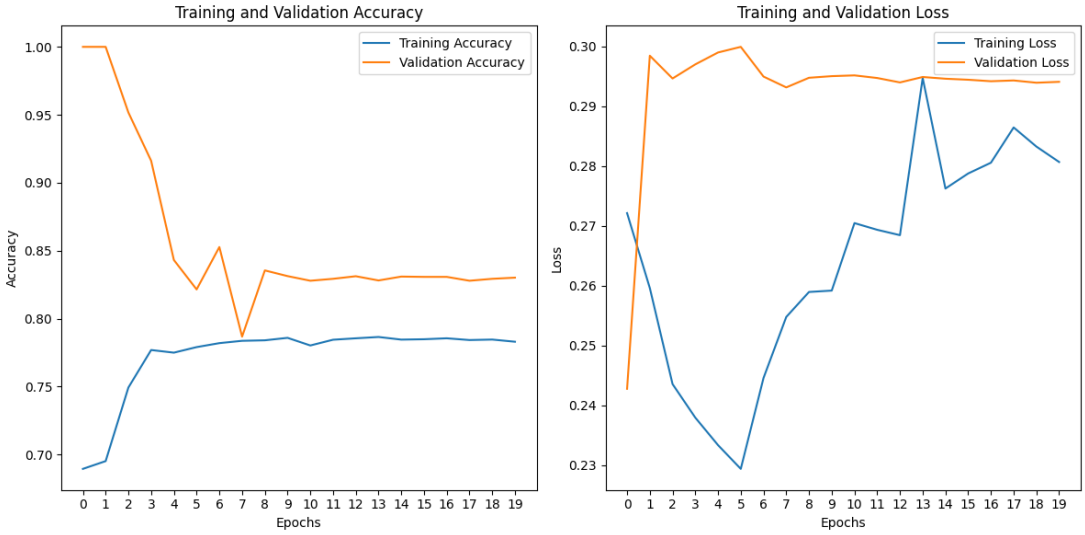
- Theo bài báo:

Epoch	$learning_{rate}$	precision	recall	F1
20	0.1	0.958	0.914	0.932
20	0.01	0.978	0.977	0.977
20	0.001	0.982	0.981	0.981
20	0.0001	0.985	0.982	0.983
20	0.00001	0.918	0.906	0.908

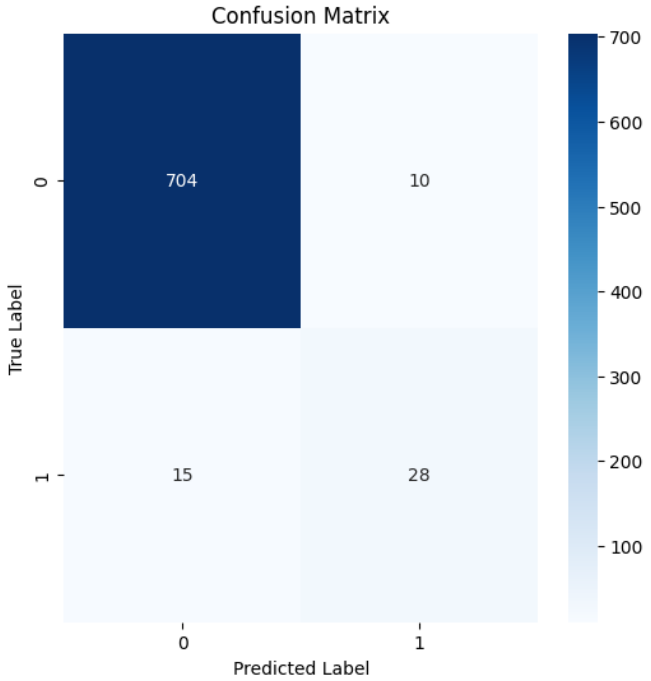
Hình 30. Kết quả demo trong bài báo.

- Kết quả demo của em:

Do khi mô hình của em khi train multiclass cho kết quả khá là thấp, nên em quyết định chuyển sang train binary class (chỉ phân loại là normal sequence transaction – 0 hoặc attack sequence transaction - 1) thì kết quả có cải tiện hơn, tuy nhiên nó cho kết quả cũng không còn cao lắm



Hình 31. Accuracy vs loss.



Hình 32. Confusion Matrix.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	714
1	0.74	0.65	0.69	43
accuracy			0.97	757
macro avg	0.86	0.82	0.84	757
weighted avg	0.97	0.97	0.97	757

Hình 33. Score.

Vì do chất lượng dataset (bằng một nửa so với bài báo) của em còn ít và không được đảm bảo chính xác hoàn nên đó là lý do chính vì sao cho kết quả thấp hơn một các rõ rệt như vậy. Hiện tại, em vẫn đang phân tích để làm tăng số lượng dataset và cải tiến mô hình .

4.2. Hướng phát triển

Mục tiêu hiện tại của em là phải cải tiến mô hình để cho kết quả tốt hơn nữa,. Còn hướng phát triển thì em sẽ tập trung theo hướng phát hiện tấn công trên các nền tảng blockchain và crosschain.

Chương 5. TỔNG KẾT, KHÓ KHĂN VÀ HẠN CHẾ

5.1. Kỹ năng học được

- Kỹ năng nghiên cứu: Nắm vững phương pháp nghiên cứu khoa học để thu thập và phân tích.
- Kỹ năng làm việc nhóm: Cải thiện khả năng giao tiếp, luôn tham gia đề xuất ý kiến để đạt được mục tiêu chung.
- Kỹ năng trình bày: Báo cáo thường xuyên giúp em tự tin hơn khi trình bày, giải thích chi tiết công việc rõ ràng hơn cho mọi người.
- Viết báo cáo khoa học: Sắp xếp báo cáo một cách có logic, từ chương mục đến chi tiết. Sử dụng từ ngữ mô tả chi tiết phương pháp và kết quả nghiên cứu.

5.2. Khó khăn

- Vì em chưa từng tiếp cận đến Blockchain trước đó, nên gặp nhiều khó khăn trong việc tìm hiểu và phân tích chuyên sâu.
- Đa phần báo cáo chi tiết liên quan đến tấn công thường viết bằng tiếng Trung Quốc, em phải sử dụng google dịch nhưng bảng dịch có nhiều câu dịch không rõ nghĩa nên em thường bị khó hiểu và nhầm lẫn.
- Thời gian gặp mặt trực tiếp giữa các thành viên trong nhóm ít vì do khác thời gian học, chủ yếu là gặp nhau online.

TÀI LIỆU THAM KHẢO

[1] Su, L., Shen, X., Du, X., Liao, X., Wang, X., Xing, L., & Liu, B. (2021). Evil under the sun: understanding and discovering attacks on Ethereum decentralized applications. In *30th USENIX Security Symposium (USENIX Security 21)* (pp. 1307-1324).

[2] Chen, T., Li, Z., Zhu, Y., Chen, J., Luo, X., Lui, J. C. S., ... & Zhang, X. (2020). Understanding ethereum via graph analysis. ACM Transactions on Internet Technology (TOIT), 20(2), 1-32.

[3] Bartoletti, T. C. N. A. M. (2016). A Survey of Attacks on Ethereum Smart Contracts. Universita degli Studi di Cagliari, Cagliari, Italy.

[4] Shen, Yun, and Gianluca Stringhini. "{ATTACK2VEC}: Leveraging Temporal Word Embeddings to Understand the Evolution of Cyberattacks." 28th USENIX Security Symposium (USENIX Security 19). 2019.

[5] Grover, Aditya and Jure Leskovec. “node2vec: Scalable Feature Learning for Networks.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)

[6] Chris Chinchilla. Ethereum white paper. https://github.com/ethereum/wiki/wiki/White-Paper. Last accessed 14/11/2023.

[7] GAVIN WOOD. Ethereum: A secure decentralised generalised transaction ledger. https://ethereum.github.io/yellowpaper/paper.pdf. Last accessed 14/11/2023.

[8] Bloxy. https://bloxy.info Last accessed 10/01/2024

[9] DappRadar. https://dappradar.com Last accessed 14/11/2024

[10] Etherscan. https://etherscan.io Last accessed 18/01/2024

[11] DASP https://dasp.co. Last accessed 23/12/2024