**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: CƠ SỞ AN TOÀN THÔNG TIN**

**MÃ HỌC PHẦN: INT1472**

**ĐỀ TÀI:  Khám phá Công nghệ Blockchain: Lý thuyết, Ứng dụng và Thử nghiệm An ninh**

Các sinh viên thực hiện

<Mã sinh viên> <Họ tên sinh viên>

Tên nhóm: 03

Tên lớp: 402

Giảng viên hướng dẫn: Đỗ Xuân Trợ

**HÀ NỘI 2025**

**PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Công việc / Nhiệm vụ** | **SV thực hiện** | **Thời hạn  hoàn thành** |
| 1 | Tìm hiểu khái quát về Moodle LMS | Nguyễn Văn A | 15/9/2024 |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

**NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **SV thực hiện** | **Thái độ tham gia** | **Mức hoàn thành CV** | **Kỹ năng giao tiếp** | **Kỹ năng hợp tác** | **Kỹ năng lãnh đạo** |
| 1 | Nguyễn Văn A | 3 | 4 | 3 | 2 | 1 |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |

***Ghi chú***:

* Thái độ tham gia: Đánh giá điểm thái độ tham gia công việc chung của nhóm (từ 0: không tham gia, đến 5: chủ động, tích cực).
* Mức hoàn thành CV: Đánh giá điểm mức độ hoàn thành công việc được giao (từ 0: không hoàn thành, đến 5: hoàn thành xuất sắc).
* Kỹ năng giao tiếp: Đánh giá điểm khả năng tương tác, giao tiếp trong nhóm (từ 0: không hoặc giao tiếp rất yếu, đến 5: giao tiếp xuất sắc).
* Kỹ năng hợp tác: Đánh giá điểm khả năng hợp tác, hỗ trợ lẫn nhau, giải quyết mâu thuẫn, xung đột
* Kỹ năng lãnh đạo: Đánh giá điểm khả năng lãnh đạo (từ 0: không có khả năng lãnh đạo, đến 5: có khả năng lãnh đạo tốt, tổ chức và điều phối công việc trong nhóm hiệu quả).

Mục lục

[Mục lục 2](#_Toc215528635)

[DANH MỤC CÁC HÌNH VẼ 5](#_Toc215528636)

[DANH MỤC CÁC BẢNG BIỂU 5](#_Toc215528637)

[DANH MỤC CÁC TỪ VIẾT TẮT 6](#_Toc215528638)

[MỞ ĐẦU 7](#_Toc215528639)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN 8](#_Toc215528640)

[1.1 Lịch sử hình thành và phát triển 8](#_Toc215528641)

[1.1.1 Sự ra đời của Bitcoin và khái niệm Blockchain đầu tiên 8](#_Toc215528642)

[1.1.2 Sự phát triển của Ethereum và Hợp đồng thông minh 8](#_Toc215528643)

[1.2 Các khái niệm và thành phần cốt lõi của Blockchain 9](#_Toc215528644)

[1.2.1 Cấu trúc khối (Block) và chuỗi (Chain) 9](#_Toc215528645)

[1.2.2 Cơ chế mã hóa và hàm băm (Hashing) 10](#_Toc215528646)

[1.2.3 Cơ chế đồng thuận (Consensus Mechanism) 10](#_Toc215528647)

[1.3 Phân loại Blockchain 11](#_Toc215528648)

[1.3.1 Blockchain Công cộng (Public Blockchain) 11](#_Toc215528649)

[1.3.2 Blockchain Riêng tư (Private Blockchain) 11](#_Toc215528650)

[1.3.3 Blockchain Liên minh (Consortium Blockchain) 11](#_Toc215528651)

[1.4 So sánh cơ sở dữ liệu và Blockchain 13](#_Toc215528652)

[1.5 Kết luận 14](#_Toc215528653)

[CHƯƠNG 2. ỨNG DỤNG VÀ TÁC ĐỘNG CỦA CÔNG NGHỆ BLOCKCHAIN 15](#_Toc215528654)

[2.1 Cách mạng trong lĩnh vực Tài chính (DeFi) 15](#_Toc215528655)

[2.1.1 Sàn giao dịch phi tập trung (DEX): 15](#_Toc215528656)

[2.1.2 Cho vay và Vay (Lending & Borrowing): 15](#_Toc215528657)

[2.1.3 Staking và Yield Farming: 16](#_Toc215528658)

[2.2 Minh bạch hóa Chuỗi cung ứng (Supply Chain) 17](#_Toc215528659)

[2.2.1 Truy xuất nguồn gốc sản phẩm: 17](#_Toc215528660)

[2.2.2 Ví dụ thực tiễn: IBM Food Trust, Vina T-check. 17](#_Toc215528661)

[2.3 Tái định hình Quyền sở hữu số 19](#_Toc215528662)

[2.3.1 NFT (Non-Fungible Token - Token không thể thay thế) trong Nghệ thuật và Sưu tầm: 19](#_Toc215528663)

[2.3.2 GameFi (Game + Finance) và nền kinh tế trong game: 20](#_Toc215528664)

[2.3.3 Định danh số (Digital Identity): 21](#_Toc215528665)

[2.4 Tiềm năng trong Quản lý công 21](#_Toc215528666)

[2.5 Kết luận 21](#_Toc215528667)

[CHƯƠNG 3. CÁC LỖ HỔNG PHỔ BIẾN TRONG HỢP ĐỒNG THÔNG MINH 21](#_Toc215528668)

[3.1 Giới thiệu về an ninh hợp đồng thông minh (smart contract) 21](#_Toc215528669)

[3.2 Lỗ hổng Tấn công Tái nhập (Reentrancy) 22](#_Toc215528670)

[3.3 Lỗ hổng Kiểm soát Truy cập bằng tx.origin 22](#_Toc215528671)

[3.4 Lỗ hổng Tấn công Từ chối Dịch vụ (Denial of Service - DoS) 23](#_Toc215528672)

[3.5 Kết luận 24](#_Toc215528673)

[CHƯƠNG 4. THỰC NGHIỆM TẤN CÔNG VÀ PHÒNG THỦ 24](#_Toc215528674)

[4.1 Giới thiệu môi trường và công cụ 24](#_Toc215528675)

[4.1.1 Ngôn ngữ lập trình: Solidity 25](#_Toc215528676)

[4.1.2 Môi trường phát triển: Remix IDE 25](#_Toc215528677)

[4.1.3 Mạng Blockchain cục bộ: Ganache 25](#_Toc215528678)

[4.2 Lab 1: Demo Tấn công Tái nhập (Reentrancy) 26](#_Toc215528679)

[4.2.1 Phân tích mã nguồn Hợp đồng có lỗ hổng (EtherStore.sol) 26](#_Toc215528680)

[4.2.2 Xây dựng mã nguồn Hợp đồng tấn công (Attack.sol) 26](#_Toc215528681)

[4.2.3 Các bước thực hiện tấn công 28](#_Toc215528682)

[4.2.4 Phân tích kết quả 28](#_Toc215528683)

[4.2.5 Vá lỗ hổng và kiểm tra lại 29](#_Toc215528684)

[4.3 Lab 2: Demo Lỗ hổng xác thực bằng tx.origin 31](#_Toc215528685)

[4.3.1 Phân tích mã nguồn Hợp đồng có lỗ hổng (Wallet.sol) 31](#_Toc215528686)

[4.3.2 Xây dựng mã nguồn Hợp đồng tấn công (AttackWallet.sol) 32](#_Toc215528687)

[4.3.3 Các bước thực hiện tấn công 33](#_Toc215528688)

[4.3.4 Phân tích kết quả 33](#_Toc215528689)

[4.3.5 Vá lỗ hổng và kiểm tra lại 36](#_Toc215528690)

[4.4 Lab 3: Demo Tấn công Từ chối Dịch vụ (DoS) 37](#_Toc215528691)

[4.4.1 Phân tích mã nguồn Hợp đồng có lỗ hổng (KingOfEther.sol) 37](#_Toc215528692)

[4.4.2 Xây dựng mã nguồn Hợp đồng tấn công (AttackDoS.sol) 38](#_Toc215528693)

[4.4.3 Các bước thực hiện tấn công 39](#_Toc215528694)

[4.4.4 Phân tích kết quả 39](#_Toc215528695)

[4.4.5 Vá lỗ hổng và kiểm tra lại 40](#_Toc215528696)

[4.5 Lab 4: Demo GameFi (Game + Finance) - Flappy Bird Blockchain 41](#_Toc215528697)

[4.5.1 Giới thiệu kịch bản ứng dụng 41](#_Toc215528698)

[4.5.2 Phân tích thiết kế Hợp đồng thông minh (Smart Contract) 42](#_Toc215528699)

[4.5.3 Tích hợp Frontend và Web3 46](#_Toc215528700)

[4.5.4 Kịch bản kiểm thử (Demo Flow) 47](#_Toc215528701)

[4.5.5 Đánh giá rủi ro bảo mật (Critical Analysis) 50](#_Toc215528702)

[4.6 Kết luận Chương 4 50](#_Toc215528703)

[CHƯƠNG 5. THÁCH THỨC, TRIỂN VỌNG VÀ KẾT LUẬN 51](#_Toc215528704)

[5.1 Những thách thức cốt lõi 51](#_Toc215528705)

[5.1.1 Thách thức về Công nghệ: "Bộ ba bất khả thi" (The Blockchain Trilemma) 51](#_Toc215528706)

[5.1.2 . Thách thức về Con người và Quản trị: Bài học từ sự sụp đổ của FTX 51](#_Toc215528707)

[5.2 Triển vọng phát triển trong tương lai 51](#_Toc215528708)

[5.2.1 Giải pháp mở rộng Layer-2: Hướng đi tất yếu 52](#_Toc215528709)

[5.2.2 Tương lai của Web3 và Metaverse 52](#_Toc215528710)

[5.3 Kết luận toàn diện 52](#_Toc215528711)

[Tài liệu tham khảo: 53](#_Toc215528712)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 9](#_Toc215528713)

[Hình 2 15](#_Toc215528714)

[Hình 3 16](#_Toc215528715)

[Hình 4 16](#_Toc215528716)

[Hình 5 17](#_Toc215528717)

[Hình 6 18](#_Toc215528718)

[Hình 7 19](#_Toc215528719)

[Hình 8 20](#_Toc215528720)

[Hình 9 20](#_Toc215528721)

[Hình 10 26](#_Toc215528722)

[Hình 11 28](#_Toc215528723)

[Hình 12 29](#_Toc215528724)

[Hình 13 30](#_Toc215528725)

[Hình 14 31](#_Toc215528726)

[Hình 15 32](#_Toc215528727)

[Hình 16 33](#_Toc215528728)

[Hình 17 34](#_Toc215528729)

[Hình 18 35](#_Toc215528730)

[Hình 19 36](#_Toc215528731)

[Hình 20 36](#_Toc215528732)

[Hình 21 37](#_Toc215528733)

[Hình 22 38](#_Toc215528734)

[Hình 23 39](#_Toc215528735)

[Hình 24 40](#_Toc215528736)

[Hình 25 41](#_Toc215528737)

[Hình 26 43](#_Toc215528738)

[Hình 27 44](#_Toc215528739)

[Hình 28 45](#_Toc215528740)

[Hình 29 46](#_Toc215528741)

[Hình 30 47](#_Toc215528742)

[Hình 31 48](#_Toc215528743)

[Hình 32 49](#_Toc215528744)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 1 13](#_Toc215528745)

[Bảng 2 14](#_Toc215528746)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ  viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| CEX | Centralized Exchange | Sàn giao dịch tập trung |
| CRUD | Create, Read, Update, Delete | Tạo, Đọc, Cập nhật, Xóa (Thao tác dữ liệu) |
| DApps | Decentralized Applications | Ứng dụng phi tập trung |
| DeFi | Decentralized Finance | Tài chính phi tập trung |
| DEX | Decentralized Exchange | Sàn giao dịch phi tập trung |
| DoS | Denial of Service | Tấn công từ chối dịch vụ |
| DPoS | Delegated Proof of Stake | Bằng chứng cổ phần ủy quyền |
| EOA | Externally Owned Account | Tài khoản sở hữu bên ngoài (Ví cá nhân) |
| ERP | Enterprise Resource Planning | Hệ thống quản lý thông tin doanh nghiệp |
| EVM | Ethereum Virtual Machine | Máy ảo Ethereum |
| GameFi | Game + Finance | Sự kết hợp giữa trò chơi và tài chính |
| IDE | Integrated Development Environment | Môi trường phát triển tích hợp |
| NFT | Non-Fungible Token | Token không thể thay thế |
| PBFT | Practical Byzantine Fault Tolerance | Khả năng chịu lỗi Byzantine thực tế |
| PoS | Proof of Stake | Bằng chứng cổ phần |
| PoW | Proof of Work | Bằng chứng công việc |
| SHA-256 | Secure Hash Algorithm 256-bit | Thuật toán băm an toàn 256-bit |
| TPS | Transactions Per Second | Số giao dịch mỗi giây |

MỞ ĐẦU

Trong bối cảnh cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ Blockchain đã và đang trở thành một trong những nền tảng có tầm ảnh hưởng sâu rộng nhất trong lĩnh vực công nghệ thông tin. Bắt đầu từ sự ra đời của Bitcoin năm 2008, Blockchain không chỉ mở ra một kỷ nguyên mới về tiền tệ số và tài chính phi tập trung (DeFi), mà còn tạo nền móng cho hàng loạt ứng dụng tiên tiến trong chuỗi cung ứng, y tế, giáo dục, quản lý dữ liệu, và đặc biệt là Web3 – Internet thế hệ mới.

Cùng với sự phát triển đó, hợp đồng thông minh (Smart Contract) – một trong những ứng dụng tiêu biểu của Blockchain đã mở rộng khả năng tự động hóa, giúp thực hiện các giao dịch và thỏa thuận mà không cần trung gian. Tuy nhiên, song song với sự tiện lợi và minh bạch của Smart Contract cũng tiềm ẩn nhiều rủi ro an ninh nghiêm trọng. Những lỗ hổng này không chỉ gây thiệt hại tài chính lớn mà còn ảnh hưởng đến niềm tin và tính ổn định của toàn bộ hệ sinh thái Blockchain.

Vì vậy, việc nghiên cứu và tìm hiểu sâu về các cơ chế hoạt động, điểm yếu bảo mật và giải pháp phòng thủ trong hợp đồng thông minh là hết sức cần thiết. Đề tài “ **Khám phá Công nghệ Blockchain: Lý thuyết, Ứng dụng và Thử nghiệm An ninh**” được lựa chọn nhằm mục tiêu:

* Hệ thống hóa cơ sở lý thuyết về công nghệ Blockchain và Smart Contract.
* Phân tích cơ chế hoạt động, mô hình triển khai và các lỗ hổng bảo mật phổ biến.
* Thực nghiệm mô phỏng tấn công - phòng thủ 3 lỗ hổng phổ biến của smart contract, qua đó rèn luyện tư duy bảo mật và nâng cao ý thức về bảo mật khi phát triển ứng dụng phi tập trung

Báo cáo được chia thành năm chương chính:

* CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN
* CHƯƠNG 2: ỨNG DỤNG VÀ TÁC ĐỘNG CỦA CÔNG NGHỆ BLOCKCHAIN
* CHƯƠNG 3: CÁC LỖ HỔNG BẢO MẬT PHỔ BIẾN TRONG HỢP ĐỒNG THÔNG MINH
* CHƯƠNG 4: THỰC NGHIỆM TẤN CÔNG VÀ PHÒNG THỦ
* CHƯƠNG 5: THÁCH THỨC, TRIỂN VỌNG VÀ KẾT LUẬN

1. TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN
   1. Lịch sử hình thành và phát triển
      1. Sự ra đời của Bitcoin và khái niệm Blockchain đầu tiên

Khái niệm Blockchain được giới thiệu lần đầu tiên vào năm 2008, thông qua bản whitepaper “Bitcoin: A Peer-to-Peer Electronic Cash System” của Satoshi Nakamoto.  
Bitcoin được thiết kế như một hệ thống thanh toán điện tử phi tập trung, cho phép các bên giao dịch trực tiếp mà không cần trung gian (ngân hàng hay tổ chức tài chính).

Công nghệ Blockchain trong Bitcoin đóng vai trò như một sổ cái phân tán (Distributed Ledger), nơi tất cả các giao dịch đều được ghi nhận trong các khối dữ liệu (block) và liên kết với nhau thành chuỗi (chain) theo thứ tự thời gian.  
Mỗi khối được xác thực thông qua cơ chế đồng thuận Proof of Work (PoW) – yêu cầu các nút mạng giải một bài toán mật mã để xác minh giao dịch, từ đó đảm bảo tính toàn vẹn và không thể chỉnh sửa dữ liệu.

Nhờ đặc tính bất biến, minh bạch và phi tập trung, Bitcoin nhanh chóng trở thành ứng dụng đầu tiên chứng minh tính khả thi của Blockchain trong thực tiễn.

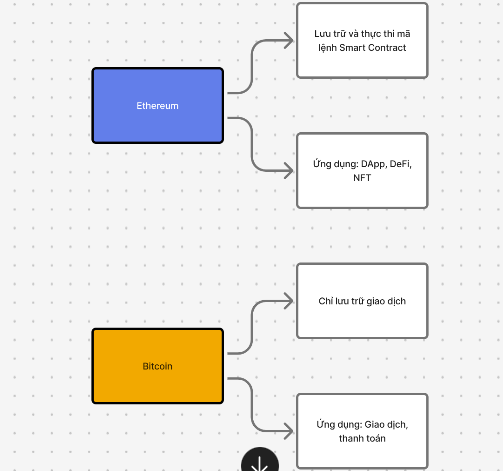
* + 1. Sự phát triển của Ethereum và Hợp đồng thông minh

Năm 2015, Ethereum ra đời – một nền tảng Blockchain thế hệ thứ hai, do Vitalik Buterin sáng lập. Nếu Bitcoin chỉ tập trung vào chuyển giá trị (giao dịch tiền tệ số), thì Ethereum mở rộng phạm vi sang tự động hóa giao dịch thông qua lập trình, với khái niệm Hợp đồng thông minh (Smart Contract).

Smart Contract là một đoạn mã chương trình được triển khai trực tiếp trên Blockchain, có khả năng tự động thực thi các điều khoản khi điều kiện được đáp ứng mà không cần sự can thiệp của con người. Điều này giúp:

* Giảm chi phí trung gian.
* Tăng tính minh bạch, vì mã lệnh và trạng thái đều công khai.
* Tăng tính an toàn, vì mọi thao tác đều được ghi nhận bất biến trên chuỗi.

Ethereum không chỉ kế thừa các ưu điểm của Bitcoin mà còn tạo nền tảng cho hàng nghìn ứng dụng phi tập trung (DApps), tài chính phi tập trung (DeFi), NFT, và Web3, góp phần mở ra kỷ nguyên Internet phi tập trung.



Hình 1

* 1. Các khái niệm và thành phần cốt lõi của Blockchain

Blockchain là sự kết hợp giữa mật mã học (cryptography), mạng máy tính phân tán (distributed network) và cơ chế đồng thuận (consensus mechanism). Ba thành phần này hợp lại tạo nên một hệ thống lưu trữ dữ liệu an toàn, minh bạch và phi tập trung.

* + 1. Cấu trúc khối (Block) và chuỗi (Chain)

Mỗi khối (block) trong Blockchain là một đơn vị lưu trữ thông tin, bao gồm hai phần chính:

* Block Header (Phần đầu khối): chứa các thông tin định danh như:

*Mã băm (hash) của khối trước đó* → đảm bảo tính liên kết chuỗi.

*Merkle Root* → mã đại diện cho toàn bộ các giao dịch trong khối.

*Timestamp* → thời gian tạo khối.

*Nonce* → giá trị dùng trong cơ chế đồng thuận (PoW).

* Block Body (Phần thân khối): chứa danh sách các giao dịch đã được xác thực.

Các khối được liên kết với nhau thông qua hàm băm (hash) tạo thành chuỗi khối (Blockchain). Một thay đổi nhỏ trong bất kỳ khối nào cũng sẽ làm thay đổi toàn bộ chuỗi sau đó, nhờ đó Blockchain đảm bảo được tính toàn vẹn và chống sửa đổi dữ liệu.

* + 1. Cơ chế mã hóa và hàm băm (Hashing)

Blockchain sử dụng hàm băm mật mã học (cryptographic hash function) – phổ biến nhất là SHA-256 (Secure Hash Algorithm 256-bit) – để tạo mã định danh duy nhất cho mỗi khối.

Đặc tính của hàm băm:

* Một chiều (One-way): không thể suy ngược lại dữ liệu ban đầu từ mã băm.
* Xác định (Deterministic): cùng một đầu vào luôn cho cùng một kết quả.
* Nhạy cảm (Sensitive): thay đổi chỉ 1 ký tự cũng dẫn đến mã băm hoàn toàn khác.
* Hiệu quả cao (Efficient): nhanh chóng xác minh dữ liệu mà không cần giải mã.

Nhờ đó, Blockchain duy trì được tính bảo mật và minh bạch tuyệt đối cho mọi giao dịch.

* + 1. Cơ chế đồng thuận (Consensus Mechanism)

Trong hệ thống phi tập trung, không có cơ quan trung gian nào kiểm soát, nên các nút mạng cần một cơ chế đồng thuận để thống nhất về tính hợp lệ của giao dịch.

Hai cơ chế phổ biến hiện nay gồm:

* Proof of Work (PoW):  
  Các nút mạng (miners) cạnh tranh giải một bài toán mật mã để xác minh khối mới. Ai giải được trước sẽ nhận phần thưởng khối. PoW mang lại bảo mật cao nhưng tiêu tốn năng lượng và tài nguyên tính toán lớn.
* Proof of Stake (PoS):  
  Các nút mạng được chọn xác thực dựa trên số lượng token mà họ “đặt cọc” (stake). PoS tiết kiệm năng lượng hơn, tăng khả năng mở rộng và được áp dụng trong Ethereum 2.0.

Ngoài ra, các biến thể như Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT) cũng được phát triển để cân bằng giữa tốc độ, bảo mật và tính phi tập trung.

* 1. Phân loại Blockchain

Dựa trên cơ chế cấp quyền và mức độ tham gia, Blockchain được chia thành ba loại chính: Public (Công cộng), Private (Riêng tư), và Consortium (Liên minh). Mỗi loại có những đặc điểm riêng về quyền truy cập, tốc độ và mức độ phi tập trung, phù hợp với các mục đích sử dụng khác nhau.

* + 1. Blockchain Công cộng (Public Blockchain)

Đây là một mạng lưới mở hoàn toàn, bất kỳ ai cũng có thể tham gia, gửi giao dịch và trở thành một nút xác thực. Public Blockchain có tính phi tập trung và bảo mật cao nhất nhờ sự tham gia của một số lượng lớn các nút mạng.

Ví dụ điển hình: Bitcoin, Ethereum.

* + 1. Blockchain Riêng tư (Private Blockchain)

Đây là một mạng lưới được kiểm soát bởi một tổ chức duy nhất. Quyền tham gia, xem dữ liệu và xác thực giao dịch bị giới hạn nghiêm ngặt. Private Blockchain thường được các doanh nghiệp sử dụng cho các ứng dụng nội bộ đòi hỏi tốc độ cao và quyền riêng tư.

Ví dụ điển hình: Hyperledger Fabric.

* + 1. Blockchain Liên minh (Consortium Blockchain)

Đây là mô hình lai giữa Public và Private, được quản lý bởi một nhóm các tổ chức thay vì một thực thể duy nhất. Consortium phù hợp cho việc hợp tác giữa các doanh nghiệp trong cùng một ngành, nơi cần sự tin tưởng và chia sẻ dữ liệu nhưng không muốn công khai hoàn toàn.

Ví dụ điển hình: R3 Corda (sử dụng trong ngành ngân hàng).

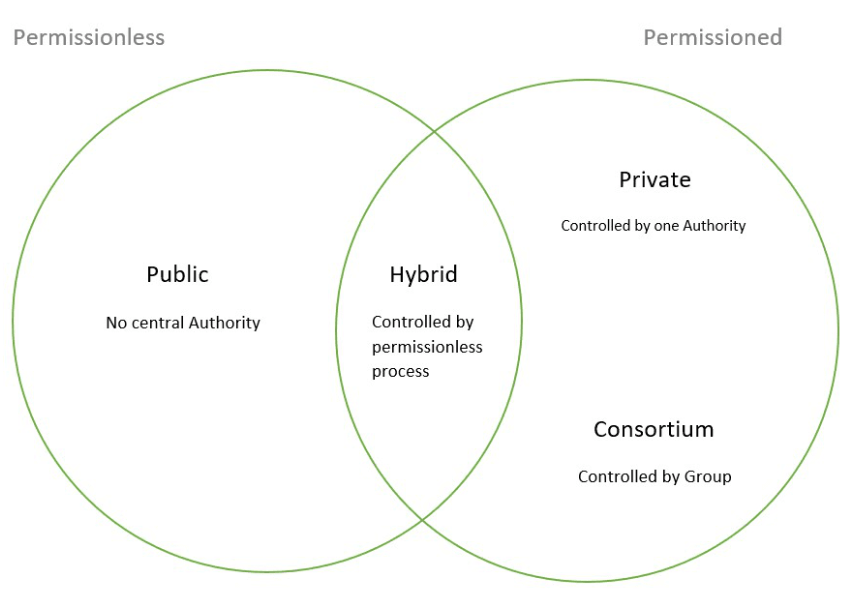
*Lưu ý: Một số tài liệu còn đề cập đến Hybrid Blockchain, là sự kết hợp giữa blockchain công cộng và riêng tư, ví dụ như sử dụng một blockchain riêng tư để xử lý giao dịch nội bộ và ghi lại bằng chứng lên blockchain công cộng để xác thực.*

Các đặc tính của ba loại Blockchain chính được so sánh chi tiết trong bảng dưới đây.

**Bảng 1.1: So sánh các loại Blockchain**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **Public Blockchain (Công cộng)** | **Private Blockchain (Riêng tư)** | **Consortium Blockchain (Liên minh)** |
| **Quyền truy cập** | Mở cho tất cả mọi người (Permissionless) | Bị giới hạn, cần cấp phép (Permissioned) | Bị giới hạn trong một nhóm (Permissioned) |
| **Người xác thực** | Bất kỳ ai | Một tổ chức duy nhất | Một nhóm tổ chức được chọn trước |
| **Mức độ phi tập trung** | Cao | Tập trung | Bán phi tập trung (Phân tán) |
| **Tốc độ giao dịch** | Chậm | Rất nhanh | Nhanh |
| **Tính bất biến** | Rất cao (Khó bị tấn công 51%) | Thấp hơn (Dễ bị kiểm soát) | Cao (Khó hơn Private) |
| **Tính minh bạch** | Cao (Tất cả giao dịch đều công khai) | Thấp (Chỉ thành viên được xem) | Trung bình (Công khai trong nhóm) |
| **Chi phí giao dịch** | Cao | Gần như bằng không | Thấp |
| **Ví dụ** | Bitcoin, Ethereum | Hyperledger Fabric | R3 Corda, Quorum |

Bảng 1



* 1. So sánh cơ sở dữ liệu và Blockchain

**Bảng 1.2: So sánh giữa Cơ sở dữ liệu truyền thống và Blockchain**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Cơ sở dữ liệu (Database)** | **Blockchain** |
| **Kiến trúc** | **Tập trung (Centralized):** Dữ liệu được lưu trữ và quản lý trên một máy chủ hoặc cụm máy chủ trung tâm. | **Phi tập trung / Phân tán (Decentralized/Distributed):** Dữ liệu được sao chép và lưu trữ trên nhiều nút mạng khác nhau. |
| **Quyền kiểm soát** | **Tập trung:** Một quản trị viên (hoặc tổ chức) có toàn quyền kiểm soát, thêm, sửa, xóa dữ liệu. | **Phân tán:** Không có cá nhân nào có quyền kiểm soát. Mọi thay đổi đều phải được mạng lưới đồng thuận. |
| **Thao tác dữ liệu (CRUD)** | Hỗ trợ đầy đủ **CRUD** (Create, Read, Update, Delete). Dữ liệu có thể được sửa đổi và xóa dễ dàng. | Về cơ bản là **chỉ ghi thêm (Append-only)**. Dữ liệu một khi đã được ghi thì không thể sửa đổi hoặc xóa (Tính bất biến). |
| **Tính toàn vẹn & Minh bạch** | Dữ liệu có thể bị thay đổi mà không để lại dấu vết. Tính minh bạch phụ thuộc vào người quản trị. | Mọi giao dịch đều được ghi lại vĩnh viễn và liên kết với nhau bằng mã hóa, đảm bảo tính toàn vẹn và minh bạch cao. |
| **Hiệu suất** | **Cao:** Tốc độ đọc/ghi nhanh, throughput lớn, tối ưu cho các truy vấn phức tạp (SQL). | **Thấp:** Tốc độ giao dịch chậm, throughput thấp do cần thời gian cho cơ chế đồng thuận và tạo khối. |
| **Khả năng chống lỗi** | **Thấp:** Dễ bị tấn công vào **điểm lỗi duy nhất (Single Point of Failure)**. Nếu máy chủ trung tâm gặp sự cố, hệ thống sẽ ngừng hoạt động. | **Cao:** Khả năng chống lỗi và chống tấn công cao. Hệ thống vẫn hoạt động ngay cả khi một vài nút mạng bị lỗi hoặc ngoại tuyến. |
| **Chi phí** | Chi phí thiết lập và bảo trì máy chủ trung tâm có thể tốn kém, nhưng chi phí giao dịch gần như bằng không. | Chi phí thiết lập ban đầu có thể thấp hơn, nhưng chi phí vận hành (phí giao dịch, năng lượng cho PoW) cao hơn. |
| **Ứng dụng phù hợp** | Hệ thống quản lý thông tin doanh nghiệp (ERP), mạng xã hội, ứng dụng web/di động cần tốc độ cao và dữ liệu tập trung. | Hệ thống cần sự tin cậy, minh bạch và không cần bên thứ ba: Tiền tệ số, chuỗi cung ứng, bỏ phiếu điện tử, quản lý tài sản số. |

Bảng 2

* 1. Kết luận

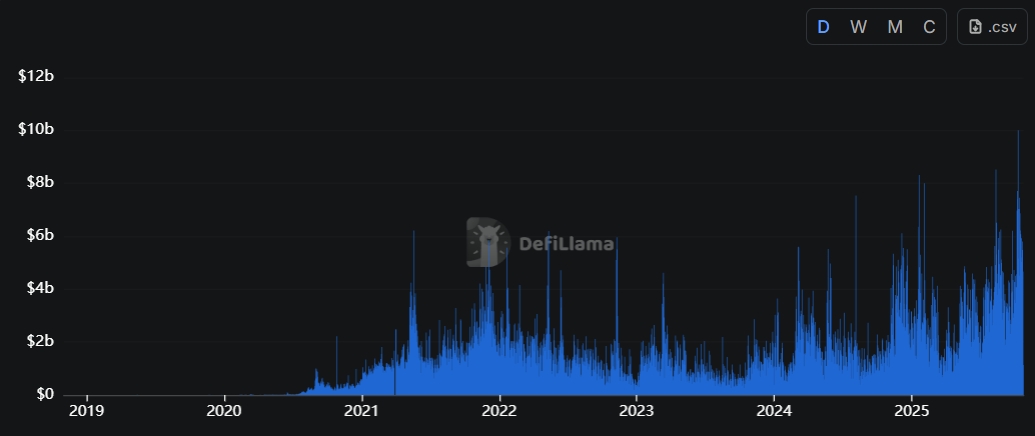
Tổng kết Chương 1, báo cáo đã hệ thống hóa những kiến thức nền tảng của công nghệ Blockchain, từ lịch sử ra đời và các thành phần kỹ thuật cốt lõi, cho đến việc phân loại và chỉ ra sự khác biệt cơ bản so với cơ sở dữ liệu truyền thống. Những nội dung này cung cấp cơ sở vững chắc để đi sâu vào phân tích các ứng dụng thực tiễn và tầm ảnh hưởng của Blockchain, vốn là nội dung chính của Chương 2.

1. ỨNG DỤNG VÀ TÁC ĐỘNG CỦA CÔNG NGHỆ BLOCKCHAIN
   1. Cách mạng trong lĩnh vực Tài chính (DeFi)

Đây là lĩnh vực ứng dụng blockchain nổi bật nhất, nhằm mục đích tạo ra một hệ thống tài chính mở, không cần qua các trung gian truyền thống như ngân hàng.

* + 1. Sàn giao dịch phi tập trung (DEX):

Là nơi cho phép người dùng có thể mua bán, trao đổi tiền mã hóa trực tiếp với nhau (ngang hàng - peer-to-peer) mà không cần qua bất kỳ bên trung gian nào như ngân hàng, sàn giao dịch tập trung (như các sàn CEX: Binance, Coinbase...) hay tổ chức tài chính.

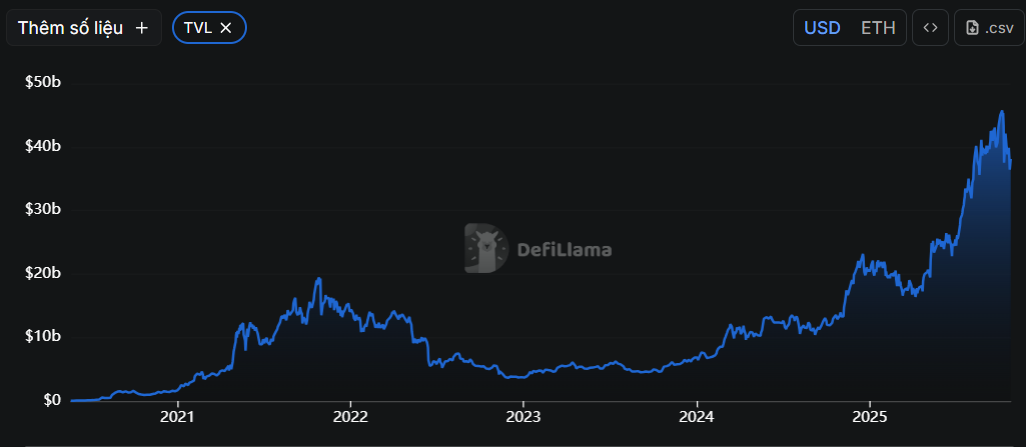
****

Hình 2

(Nguồn: [Uniswap - DefiLlama](https://defillama.com/protocol/dexs/uniswap))

* + 1. Cho vay và Vay (Lending & Borrowing):

Các nền tảng (giao thức) cho phép bạn gửi tài sản mã hóa của mình vào để cho người khác vay và nhận lãi suất, hoặc bạn có thể thế chấp tài sản của mình để vay một tài sản khác không cần thông qua các tổ chức trung gian như ngân hàng hay công ty tín dụng.

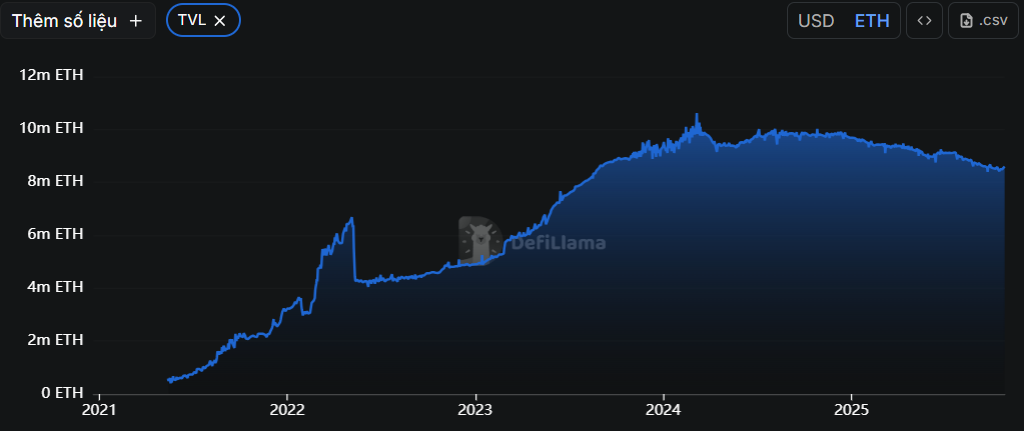
****

Hình 3

(Nguồn: [Aave - DefiLlama](https://defillama.com/protocol/aave?events=false))

* + 1. Staking và Yield Farming:

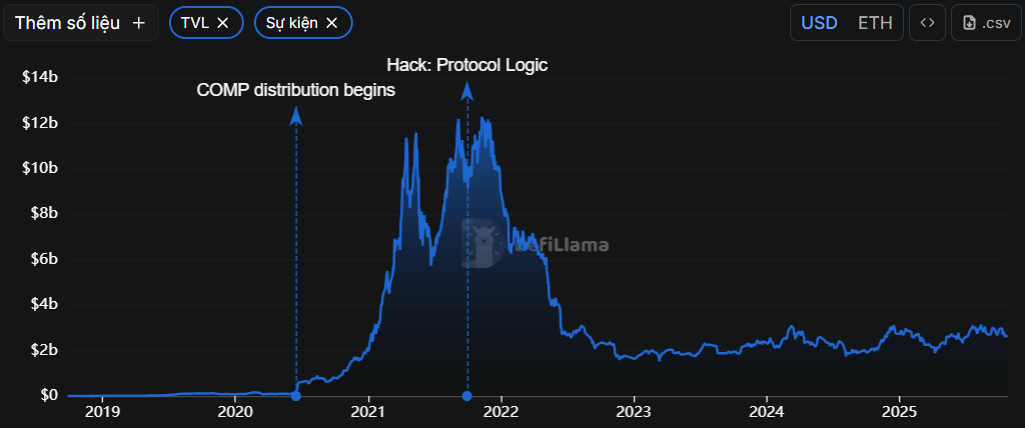
Staking: Bạn "khóa" đồng coin của mình lại để hỗ trợ bảo mật và vận hành mạng lưới (thường là trên các blockchain Proof-of-Stake) và nhận lại phần thưởng là chính đồng coin đó.

****

Hình 4

(Nguồn: [Lido - DefiLlama](https://defillama.com/protocol/lido?events=false&denomination=ETH))

**Yield Farming:** nơi người dùng liên tục di chuyển tài sản của họ qua nhiều giao thức DeFi khác nhau (cho vay, cung cấp thanh khoản...) để tối đa hóa lợi nhuận.



Hình 5

(Nguồn: [Compound Finance - DefiLlama](https://defillama.com/protocol/compound-finance))

* 1. Minh bạch hóa Chuỗi cung ứng (Supply Chain)

Blockchain cung cấp một sổ cái phân tán, công khai và không thể thay đổi, khiến nó trở nên lý tưởng cho việc theo dõi hàng hóa.

* + 1. Truy xuất nguồn gốc sản phẩm:

Mọi bước trong quá trình sản xuất và vận chuyển (từ trang trại, nhà máy, kho bãi, đến siêu thị) có thể được ghi lại trên blockchain. Người tiêu dùng cuối chỉ cần quét mã QR là có thể xem toàn bộ lịch sử của sản phẩm, đảm bảo tính xác thực và chống hàng giả (ví dụ: thực phẩm sạch, dược phẩm, hàng xa xỉ).

* + 1. Ví dụ thực tiễn: IBM Food Trust, Vina T-check.

IBM Food Trust: là một mạng lưới cộng tác bao gồm người nuôi trồng, chế biến, các nhà bán sỉ – lẻ, nhà phân phối, sản xuất, v.v. nhằm nâng cao độ nhận diện và tin cậy xuyên suốt chuỗi cung ứng thực phẩm. Được xây dựng trên nền tảng IBM Blockchain, giải pháp này không những cung cấp một quy trình hoạt động hiệu quả, mà còn mang lại lợi ích cho tất cả những người tham gia mạng lưới qua một hệ sinh thái thực phẩm an toàn, thông minh và bền vững hơn bằng cách số hóa tất cả các giao dịch cũng như dữ liệu.



Hình 6

(Hình ảnh mô tả hành trình của một con cá từ lúc đánh bắt đến khi lên đĩa ăn của bạn, mỗi bước đều được ghi lại trên sổ cái blockchain – Nguồn: Khám phá IBM - IBM Blockchain)

Vina T-check: là một giải pháp của Việt Nam tập trung vào việc chống hàng giả, hàng nhái và truy xuất nguồn gốc sản phẩm, đặc biệt cho các mặt hàng như nông sản, dược phẩm, mỹ phẩm.

Ví dụ: Tính năng nổi bật của phần mềm chống hàng giả Vinacheck: Giải pháp phần mềm chống hàng giả Vinacheck được tích hợp với tem chống giả, cùng với các công nghệ chống hàng giả như xác thực SMS, công nghệ truy xuất nguồn gốc QR Code. giúp doanh nghiệp có công cụ quản trị doanh nghiệp, quản trị lưu lượng hàng hóa đang lưu thông trên thị trường, chống bán lấn tuyến, lấn vùng; hỗ trợ quản lý hệ thống bán hàng, chăm sóc khách hàng, quản lý chương trình khuyến mãi, truy vết đường đi sản phẩm, quản lý kho mang tính hệ thống chuyên nghiệp từ khâu sản xuất đến các đại lý, chăm sóc khách hàng với nhiều cấp phân quyền.



Hình 7

(Nguồn: Giải pháp phần mềm chống hàng giả trên nền tảng số Vinacheck)

* 1. Tái định hình Quyền sở hữu số

Blockchain cho phép chứng minh quyền sở hữu đối với tài sản kỹ thuật số (và cả tài sản vật lý được "mã hóa") một cách rõ ràng.

* + 1. NFT (Non-Fungible Token - Token không thể thay thế) trong Nghệ thuật và Sưu tầm:

Là một chứng chỉ kỹ thuật số duy nhất, chứng minh bạn là chủ sở hữu của một tệp kỹ thuật số (như tranh ảnh, âm nhạc, video...). Nó giải quyết vấn đề "sao chép" vô hạn trên Internet.



Hình 8

(Tác phẩm nghệ thuật kỹ thuật số "Everydays: The First 5000 Days"được bán tại **Christie’s** vào tháng 3/2021 với giá **69,3 triệu USD**)

* + 1. GameFi (Game + Finance) và nền kinh tế trong game:

Trong các game blockchain, người chơi có thể thực sự "sở hữu" các vật phẩm (như vũ khí, trang phục) dưới dạng NFT. Họ có thể tự do mua bán, trao đổi chúng trên các khu chợ và thậm chí kiếm được tiền thật từ việc chơi game (Play-to-Earn).



Hình 9

(Biểu đồ giá của token $AXS (Axie Infinity Shard) của gamefi **Axie Infinity** )

* + 1. Định danh số (Digital Identity):

Sử dụng blockchain để tạo ra một hệ thống danh tính kỹ thuật số an toàn, do chính người dùng kiểm soát. Thay vì để các công ty (như Google, Facebook) nắm giữ dữ liệu của bạn, bạn sẽ tự quản lý "danh tính" của mình và chọn chia sẻ thông tin nào cho ai.

* 1. Tiềm năng trong Quản lý công

Đây là lĩnh vực khám phá cách các cơ quan chính phủ có thể sử dụng blockchain để tăng tính minh bạch, hiệu quả và giảm tham nhũng.

Ví dụ: Estonia là quốc gia đi đầu. Gần như toàn bộ dịch vụ công của họ (ngoại trừ kết hôn, ly hôn và giao dịch bất động sản) đều được số hóa. Hệ thống của họ sử dụng công nghệ (tương tự blockchain) tên là KSI Blockchain để đảm bảo tính toàn vẹn của dữ liệu công dân, hồ sơ y tế, và hồ sơ pháp lý, khiến chúng không thể bị can thiệp bởi bất kỳ ai (kể cả chính phủ).. Hệ thống này giúp Estonia tiết kiệm ước tính 2% GDP mỗi năm (tương đương toàn bộ ngân sách quốc phòng) nhờ giảm thiểu giấy tờ và gian lận. (Nguồn: Story - e-Estonia)

* 1. Kết luận

Chương 2 đã cho thấy một bức tranh rõ ràng: công nghệ blockchain không còn là lý thuyết mà đã trở thành một nền kinh tế kỹ thuật số thực thụ, đang lưu trữ một giá trị khổng lồ. Khi giá trị tài sản trên Blockchain ngày càng lớn, thì an ninh lại trở thành yếu tố sống còn.

1. CÁC LỖ HỔNG PHỔ BIẾN TRONG HỢP ĐỒNG THÔNG MINH
   1. Giới thiệu về an ninh hợp đồng thông minh (smart contract)

Chương 2 đã cho thấy một bức tranh toàn cảnh về giá trị khổng lồ mà các ứng dụng phi tập trung đang quản lý, từ hàng tỷ đô la trong các giao thức DeFi đến các tài sản số độc nhất (NFT). Khi giá trị càng lớn, Hợp đồng thông minh càng trở thành mục tiêu hấp dẫn cho tin tặc.

Về bản chất, Hợp đồng thông minh chỉ là những đoạn mã được thực thi tự động. Bất kỳ một sai sót logic hay một sơ hở nhỏ nào trong mã lệnh cũng có thể bị khai thác, dẫn đến những hậu quả tài chính thảm khốc và **không thể đảo ngược** do tính bất biến của Blockchain. Lịch sử đã chứng kiến nhiều vụ tấn công gây thiệt hại hàng trăm triệu đô la, làm lung lay niềm tin của cả một hệ sinh thái.

Vì vậy, việc hiểu rõ các rủi ro và trang bị tư duy "phòng thủ trước khi tấn công" là yêu cầu bắt buộc đối với bất kỳ nhà phát triển nào. Chương này sẽ tập trung phân tích sâu về **lý thuyết của 3 lỗ hổng bảo mật phổ biến và nguy hiểm nhất**, đồng thời trình bày các kỹ thuật phòng thủ tiêu chuẩn đã được cộng đồng công nhận, làm nền tảng cho phần thực nghiệm ở chương sau.

* 1. Lỗ hổng Tấn công Tái nhập (Reentrancy)

Đây là một trong những lỗ hổng khét tiếng nhất, là nguyên nhân gây ra vụ hack The DAO năm 2016, đánh cắp 60 triệu USD giá trị Ether và dẫn đến một cuộc hard-fork chia tách Ethereum thành hai chuỗi riêng biệt.

**Mô tả Lỗ hổng:** Lỗ hổng xảy ra khi một hợp đồng thực hiện một lệnh gọi ra bên ngoài (ví dụ: chuyển tiền) **trước khi** nó cập nhật trạng thái nội bộ của mình (ví dụ: trừ số dư).

**Cơ chế Tấn công:**

1. Hợp đồng của Kẻ tấn công (A) gọi hàm rút tiền trên Hợp đồng Nạn nhân (B).
2. Hợp đồng Nạn nhân (B) kiểm tra số dư và gửi Ether cho Hợp đồng (A).
3. Hợp đồng (A) nhận được Ether, hàm receive() (hoặc fallback()) của nó được kích hoạt. Thay vì kết thúc, hàm này ngay lập tức gọi **ngược trở lại** (re-enter) hàm rút tiền của Hợp đồng (B).
4. Vì Hợp đồng (B) **chưa kịp cập nhật số dư** của (A) về 0, nó lại thấy (A) vẫn còn tiền và tiếp tục gửi Ether.
5. Quá trình này lặp lại cho đến khi toàn bộ Ether trong Hợp đồng Nạn nhân bị rút cạn.

**Kỹ thuật Phòng thủ: Mẫu thiết kế "Checks-Effects-Interactions"**Đây là nguyên tắc vàng để chống lại Reentrancy. Một hàm phải luôn thực hiện theo thứ tự:

* 1. **Checks (Kiểm tra):** Kiểm tra tất cả các điều kiện (ví dụ: require(msg.sender có đủ số dư không?).
  2. **Effects (Tác động):** Thay đổi trạng thái của hợp đồng (ví dụ: cập nhật số dư của msg.sender về 0).
  3. **Interactions (Tương tác):** Thực hiện lệnh gọi ra các hợp đồng bên ngoài (ví dụ: chuyển Ether).

Bằng cách cập nhật trạng thái **trước khi** gửi tiền, kể cả khi kẻ tấn công gọi ngược lại, hợp đồng sẽ thấy số dư đã bằng 0 và từ chối giao dịch.

* 1. Lỗ hổng Kiểm soát Truy cập bằng tx.origin

**Mô tả Lỗ hổng:** Lỗ hổng xảy ra khi hợp đồng sử dụng biến toàn cục tx.origin để xác thực quyền của người dùng, thay vì msg.sender.

Chú thích

* 1. tx.origin: Là địa chỉ ví (EOA) đã **khởi tạo** giao dịch gốc.
  2. msg.sender: Là địa chỉ đã **gọi trực tiếp** hàm hiện tại (có thể là một ví hoặc một hợp đồng khác).

**Cơ chế Tấn công (Tấn công lừa đảo - Phishing):**

* 1. Kẻ tấn công tạo ra một Hợp đồng Tấn công (A) và lừa Chủ sở hữu (C) của Hợp đồng Ví (B) tương tác với nó (ví dụ: dụ dỗ nhận airdrop miễn phí).
  2. Chủ sở hữu (C) gửi một giao dịch nhỏ đến Hợp đồng Tấn công (A).
  3. Hợp đồng (A) nhận được giao dịch và ngay lập tức gọi hàm transfer() trên Hợp đồng Ví (B) để rút toàn bộ tài sản về ví của Kẻ tấn công.
  4. Bên trong Hợp đồng Ví (B), câu lệnh require(tx.origin == owner) được kiểm tra. Vì giao dịch **gốc** được khởi tạo bởi Chủ sở hữu (C), nên tx.origin chính là địa chỉ của (C). Câu lệnh này **đúng** và giao dịch được thực thi, tiền bị đánh cắp.

**Kỹ thuật Phòng thủ: Luôn sử dụng**

Để phòng thủ, chỉ cần thay tx.origin bằng msg.sender.  
Trong kịch bản tấn công trên, khi Hợp đồng (A) gọi Hợp đồng (B), msg.sender sẽ là địa chỉ của Hợp đồng (A), không phải địa chỉ của Chủ sở hữu (C). Do đó, câu lệnh require(msg.sender == owner) sẽ **sai** và giao dịch bị hủy, bảo vệ an toàn cho tài sản.

* 1. Lỗ hổng Tấn công Từ chối Dịch vụ (Denial of Service - DoS)

**Mô tả Lỗ hổng:** Trong bối cảnh Smart Contract, DoS xảy ra khi một chức năng quan trọng của hợp đồng bị "đóng băng", không ai có thể sử dụng được nữa. Một nguyên nhân phổ biến là do vòng lặp duyệt qua một mảng dữ liệu quá lớn, làm tiêu tốn hết gas.

**Cơ chế Tấn công (Qua vòng lặp):**

* 1. Một hợp đồng có chức năng phân phát thưởng cho một danh sách người đóng góp, được lưu trong một mảng địa chỉ. Hàm này sẽ lặp qua toàn bộ mảng để gửi tiền cho từng người.
  2. Khi danh sách còn nhỏ, hàm hoạt động bình thường. Nhưng khi danh sách ngày càng lớn, tổng chi phí Gas để thực hiện vòng lặp sẽ tăng lên.
  3. Nếu tổng Gas vượt quá giới hạn Gas của một block (Block Gas Limit), không một thợ đào nào có thể xử lý giao dịch đó. Toàn bộ hàm phân phát thưởng sẽ bị thất bại vĩnh viễn, không ai có thể nhận được tiền của mình.
  4. Kẻ tấn công có thể đẩy nhanh quá trình này bằng cách cố tình thêm nhiều địa chỉ rác vào danh sách (nếu có thể).

**Kỹ thuật Phòng thủ: Mẫu thiết kế "Pull-over-Push" (Kéo thay vì Đẩy)**Nguyên tắc là thay đổi logic: thay vì hợp đồng **đẩy** tiền đến cho người dùng, hãy để người dùng tự **kéo** tiền về.

1. Hợp đồng sẽ lưu lại số tiền mỗi người dùng được quyền nhận (ví dụ: trong một mapping).
2. Tạo một hàm claimReward() (hoặc withdraw()) để người dùng tự gọi.
3. Hàm này chỉ xử lý việc chuyển tiền cho một người (msg.sender) tại một thời điểm. Điều này loại bỏ hoàn toàn vòng lặp tốn gas. Mỗi người sẽ tự trả gas cho giao dịch rút tiền của chính họ.
   1. Kết luận

Chương này đã đi sâu phân tích ba trong số các lỗ hổng bảo mật nghiêm trọng nhất của Hợp đồng thông minh: Tấn công Tái nhập, lạm dụng tx.origin, và Tấn công Từ chối Dịch vụ. Qua đó, ta thấy rằng việc xây dựng ứng dụng phi tập trung an toàn đòi hỏi một tư duy lập trình cẩn trọng và tuân thủ nghiêm ngặt các mẫu thiết kế đã được chứng minh như "Checks-Effects-Interactions" và "Pull-over-Push".

Việc hiểu rõ các rủi ro này về mặt lý thuyết chính là bước đệm cần thiết để tiến vào **Chương 4**, nơi chúng ta sẽ trực tiếp thực nghiệm, mô phỏng lại các kịch bản tấn công và áp dụng các biện pháp phòng thủ này trong môi trường thực tế.

1. THỰC NGHIỆM TẤN CÔNG VÀ PHÒNG THỦ
   1. Giới thiệu môi trường và công cụ

Để tiến hành các thử nghiệm tấn công và phòng thủ một cách trực quan và hiệu quả, nhóm đã lựa chọn và sử dụng một bộ công cụ tiêu chuẩn trong ngành phát triển ứng dụng phi tập trung trên nền tảng Ethereum. Sự kết hợp của các công cụ này tạo ra một môi trường hoàn chỉnh từ khâu viết mã, triển khai cho đến kiểm thử bảo mật.

* + 1. Ngôn ngữ lập trình: Solidity
* **Vai trò:** Solidity là ngôn ngữ lập trình bậc cao, hướng đối tượng, được thiết kế đặc biệt để viết Hợp đồng thông minh trên các blockchain tương thích với Máy ảo Ethereum (EVM).
* **Lý do lựa chọn:** Đây là ngôn ngữ phổ biến nhất với cộng đồng hỗ trợ lớn mạnh và hệ sinh thái phong phú. Việc sử dụng Solidity là lựa chọn tất yếu để mô phỏng lại các lỗ hổng kinh điển như Tấn công Tái nhập hay lạm dụng tx.origin một cách chính xác nhất.
  + 1. Môi trường phát triển: Remix IDE
* **Vai trò:** Remix là một Môi trường phát triển tích hợp (IDE) mạnh mẽ, hoạt động trực tiếp trên trình duyệt web mà không yêu cầu cài đặt phức tạp.
* **Lý do lựa chọn:** Nhóm lựa chọn Remix vì sự tiện lợi và bộ công cụ "tất cả trong một". Remix tích hợp sẵn trình biên dịch, trình gỡ lỗi (debugger), và giao diện đồ họa để triển khai và tương tác trực tiếp với các hàm của Hợp đồng thông minh. Điều này giúp quá trình demo các kịch bản tấn công và phòng thủ trở nên rất rõ ràng và dễ theo dõi.
  + 1. Mạng Blockchain cục bộ: Ganache

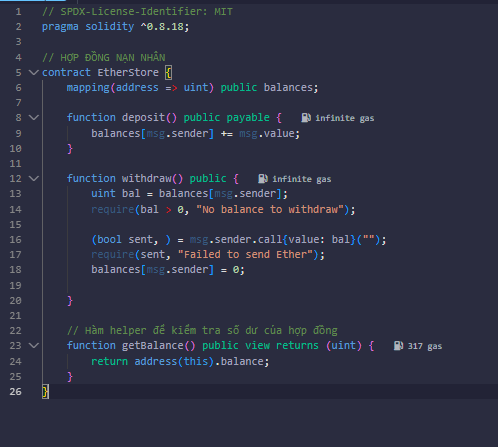
**Vai trò:** Ganache là một công cụ tạo ra một mạng Blockchain Ethereum cá nhân, chạy cục bộ trên máy tính của nhà phát triển. Nó mô phỏng lại hoạt động của mạng Ethereum thật trong một môi trường biệt lập.

**Lý do lựa chọn:** Việc sử dụng Ganache mang lại ba lợi ích quan trọng cho đề tài:

* **Tốc độ:** Giao dịch được xác nhận ngay lập tức, giúp tiết kiệm tối đa thời gian gỡ lỗi và kiểm thử.
* **Chi phí bằng không:** Ganache cung cấp sẵn các tài khoản thử nghiệm với 100 ETH, loại bỏ nhu cầu sử dụng tiền thật hay xin ETH từ các mạng thử nghiệm công cộng (Testnet).
* **Môi trường an toàn:** Cho phép thực hiện các cuộc tấn công mô phỏng trong một "hộp cát" (sandbox), không gây ảnh hưởng hay tốn kém chi phí trên mạng lưới thật.
  1. Lab 1: Demo Tấn công Tái nhập (Reentrancy)
     1. Phân tích mã nguồn Hợp đồng có lỗ hổng (EtherStore.sol)

Hợp đồng EtherStore được thiết kế như một két sắt cho phép người dùng gửi (deposit) và rút (withdraw) Ether. Tuy nhiên, trong hàm withdraw(), quy trình xử lý vi phạm nguyên tắc an toàn:

* **Lỗi:** Hợp đồng thực hiện việc chuyển Ether trả lại cho người dùng msg.sender.call{value: bal}("") **trước khi** cập nhật số dư của người đó về 0 balances[msg.sender] = 0.
* **Hậu quả:** Điều này tạo ra một cửa sổ thời gian cho phép kẻ tấn công gọi đệ quy lại hàm withdraw nhiều lần trước khi số dư được cập nhật, từ đó rút vượt quá số tiền họ sở hữu.

****

Hình 10

* + 1. Xây dựng mã nguồn Hợp đồng tấn công (Attack.sol)

Hợp đồng Attack được thiết kế để khai thác lỗ hổng trên thông qua cơ chế fallback/receive:

* Hàm attack(): Nạp một lượng nhỏ Ether vào EtherStore và gọi lệnh rút tiền đầu tiên.
* Hàm receive(): Khi nhận được Ether từ EtherStore, hàm này tự động kích hoạt. Nó kiểm tra xem EtherStore còn tiền hay không; nếu còn, nó lập tức gọi lại hàm withdraw() của EtherStore. Vòng lặp này tiếp diễn cho đến khi nạn nhân cạn kiệt tài sản.

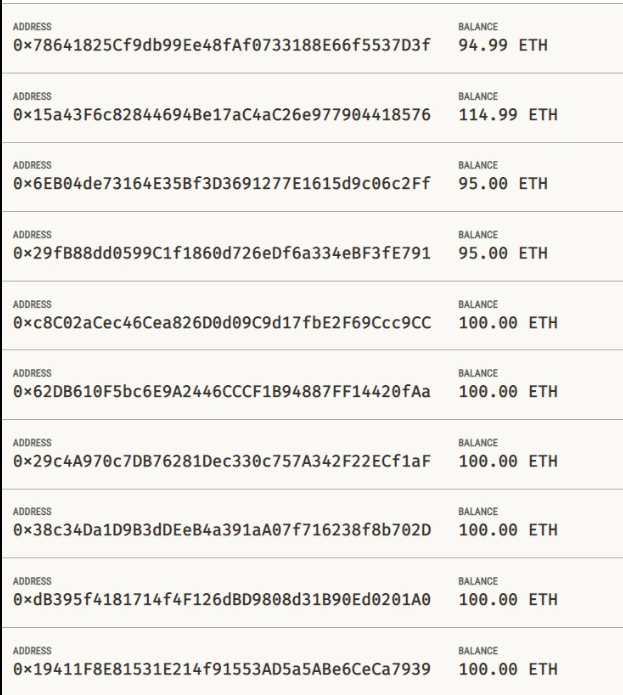
****

Hình 11

* + 1. Các bước thực hiện tấn công

1. **Triển khai (Deploy):** Sử dụng Account 1 để deploy EtherStore.
2. **Nạp tiền (Deposit):** Sử dụng Account 1,2 và Account 3 nạp tổng cộng 15 ETH vào EtherStore (mỗi người 5 ETH).
3. **Triển khai kẻ tấn công:** Sử dụng Account 4 (Hacker) để deploy hợp đồng Attack với tham số là địa chỉ của EtherStore.
4. **Tấn công:** Hacker gọi hàm attack() với giá trị gửi kèm là 1 ETH.
   * 1. Phân tích kết quả

* **Trước tấn công:** EtherStore có 15 ETH. Hợp đồng Attack có 0 ETH.
* **Sau tấn công:**
  + Số dư của EtherStore về 0.
  + Số dư của Attack tăng lên 16 ETH (15 ETH lấy cắp + 1 ETH vốn gốc).
* **Kết luận:** Cuộc tấn công thành công do logic cập nhật trạng thái bị thực hiện chậm trễ.

****

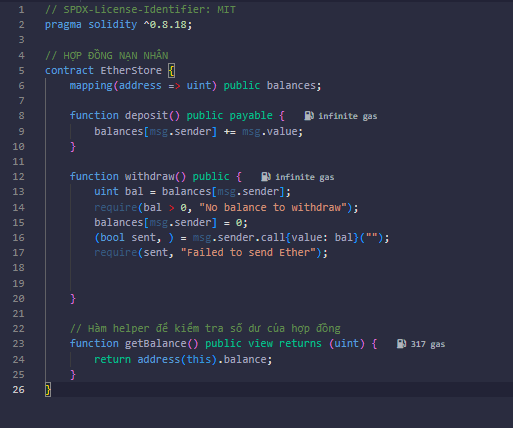
Hình 12

* + 1. Vá lỗ hổng và kiểm tra lại

a) Giải pháp khắc phục

Để khắc phục lỗ hổng Reentrancy, chúng ta áp dụng mẫu thiết kế bảo mật tiêu chuẩn "Checks-Effects-Interactions". Nguyên tắc cốt lõi là thay đổi thứ tự thực thi trong hàm withdraw(): phải cập nhật trạng thái nội bộ của hợp đồng (trừ số dư về 0) trước khi thực hiện bất kỳ tương tác nào với bên ngoài (gửi Ether).

* Mã nguồn đã vá lỗi

****

Hình 13

b) Kiểm tra lại (Re-test)

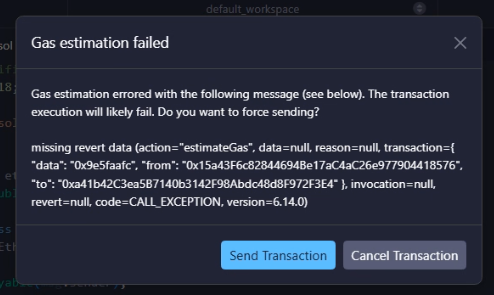
Nhóm tiến hành thực nghiệm lại kịch bản tấn công với mã nguồn đã vá:

1. Triển khai lại: Deploy EtherStore (bản mới) và nạp vào 5 ETH.
2. Chuẩn bị tấn công: Deploy lại hợp đồng Attack trỏ tới địa chỉ EtherStore mới.
3. Thực hiện tấn công: Hacker gọi hàm attack() kèm 1 ETH.

c) Kết quả

Cuộc tấn công thất bại hoàn toàn. Giao dịch bị Blockchain từ chối (Revert) ngay lập tức.

* Nguyên nhân: Khi hợp đồng Attack nhận tiền và cố gắng gọi lại withdraw() lần thứ hai (re-enter), dòng lệnh balances[msg.sender] = 0 đã được thực thi trước đó. Do vậy, điều kiện kiểm tra require(bal > 0) bị sai, dẫn đến việc toàn bộ giao dịch bị hủy bỏ để bảo toàn tài sản.

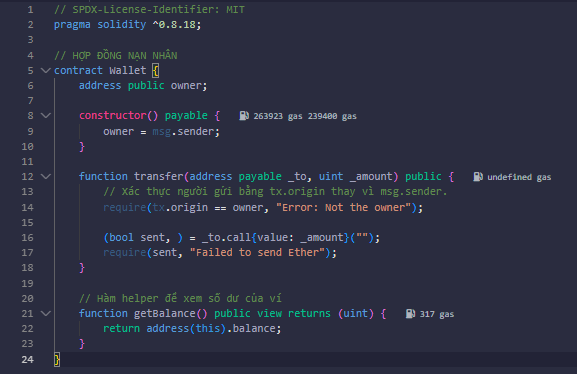
****

Hình 14

* 1. Lab 2: Demo Lỗ hổng xác thực bằng tx.origin
     1. Phân tích mã nguồn Hợp đồng có lỗ hổng (Wallet.sol)

Hợp đồng Wallet mô phỏng ví cá nhân, có chức năng chuyển tiền transfer().

* Lỗi: Hợp đồng sử dụng tx.origin để xác thực chủ sở hữu: require(tx.origin == owner).
* Nguyên nhân: tx.origin chỉ xác định người khởi tạo giao dịch ban đầu, không xác định người gọi trực tiếp. Điều này cho phép một hợp đồng trung gian mạo danh chủ sở hữu.

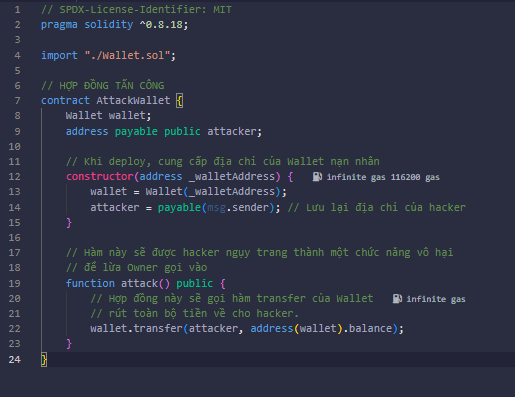
****

Hình 15

* + 1. Xây dựng mã nguồn Hợp đồng tấn công (AttackWallet.sol)

Hợp đồng AttackWallet đóng vai trò là bẫy phishing.

* Nó chứa một hàm attack() mà bên trong đó gọi đến wallet.transfer().
* Kẻ tấn công lừa chủ sở hữu ví gọi hàm attack() này. Khi đó, tx.origin là chủ sở hữu (hợp lệ), nhưng msg.sender là hợp đồng tấn công. Do dùng tx.origin, hợp đồng Wallet lầm tưởng yêu cầu là hợp lệ và chuyển tiền đi.

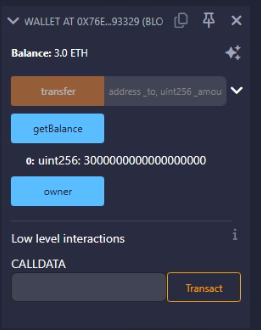
****

Hình 16

* + 1. Các bước thực hiện tấn công

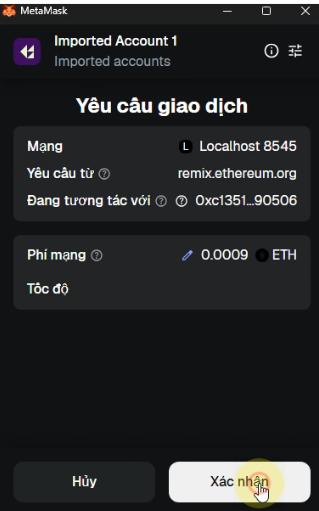
1. Deploy: Chủ ví (Account 1) deploy Wallet và nạp 3 ETH.
2. Deploy Attack: Hacker (Account 2) deploy AttackWallet.
3. Lừa đảo (Phishing): Hacker dụ dỗ Chủ ví (Account 1) tương tác và gọi hàm attack() trên hợp đồng AttackWallet.
   * 1. Phân tích kết quả

* Balance Wallet lúc đầu

****

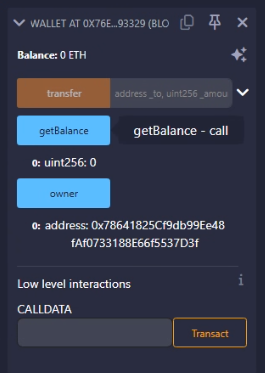
Hình 17

* Ngay khi Chủ ví xác nhận giao dịch trên hợp đồng AttackWallet, toàn bộ 3 ETH trong Wallet bị chuyển sang ví của Hacker.

****

Hình 18

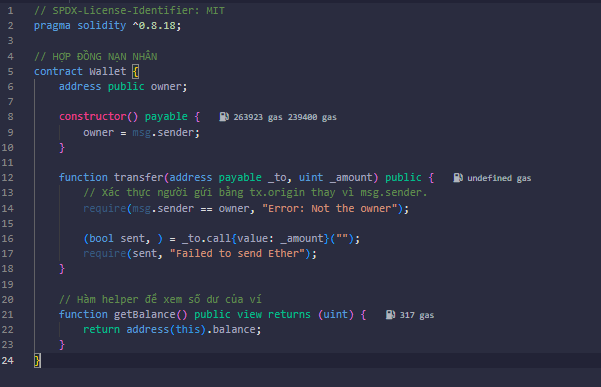
* Balance Wallet sau khi chủ Wallet bị lừa

****

Hình 19

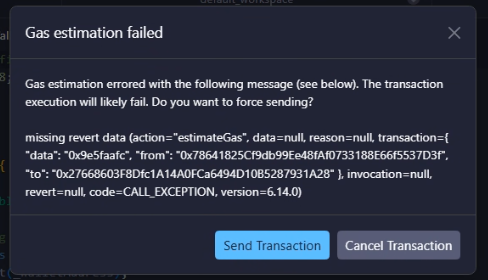
* + 1. Vá lỗ hổng và kiểm tra lại

Thay thế điều kiện xác thực tx.origin bằng msg.sender: require(msg.sender == owner).

****

Hình 20

msg.sender đại diện cho người gọi trực tiếp. Khi Hacker dùng hợp đồng trung gian, msg.sender sẽ là địa chỉ hợp đồng tấn công (không phải owner), do đó giao dịch bị từ chối.

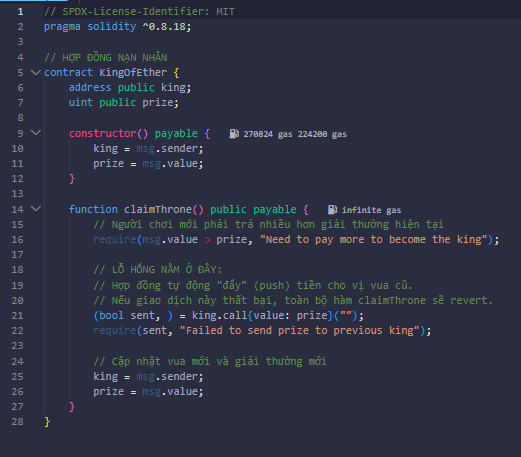
****

Hình 21

* 1. Lab 3: Demo Tấn công Từ chối Dịch vụ (DoS)
     1. Phân tích mã nguồn Hợp đồng có lỗ hổng (KingOfEther.sol)

Trò chơi KingOfEther cho phép người chơi cướp ngôi vua bằng cách gửi số tiền lớn hơn vua hiện tại. Hợp đồng sẽ tự động hoàn trả tiền cho vua cũ ngay trong giao dịch đó.

* Lỗi: Sử dụng cơ chế "Push" (Đẩy tiền): king.call{value: prize}(""). Nếu việc gửi tiền thất bại, toàn bộ giao dịch cướp ngôi sẽ thất bại (revert).

****

Hình 22

* + 1. Xây dựng mã nguồn Hợp đồng tấn công (AttackDoS.sol)

Kẻ tấn công tạo một hợp đồng với mục đích làm "Vua lỳ lợm".

* Hàm receive() của hợp đồng này chứa lệnh revert().
* Điều này có nghĩa là hợp đồng này từ chối nhận bất kỳ khoản Ether nào.

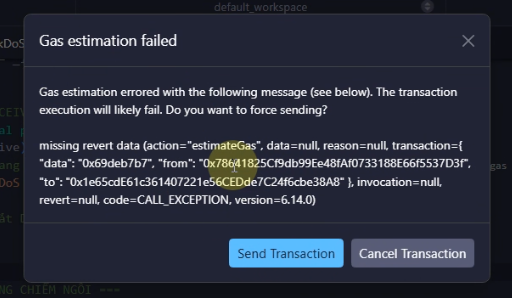
****

Hình 23

* + 1. Các bước thực hiện tấn công

1. Deploy KingOfEther. Người chơi A trở thành Vua với 1 ETH.
2. Hacker dùng hợp đồng AttackDoS gửi 2 ETH để trở thành Vua mới.
3. Người chơi B cố gắng gửi 3 ETH để chiếm ngôi từ Hacker.
   * 1. Phân tích kết quả

* Khi Người chơi B cố gắng chiếm ngôi, KingOfEther cố gửi trả 2 ETH cho AttackDoS.
* Hàm receive() của AttackDoS kích hoạt revert, làm giao dịch hoàn tiền thất bại.
* Kết quả: Giao dịch của Người chơi B bị hủy bỏ. Không ai có thể trở thành Vua mới. Hợp đồng bị tê liệt vĩnh viễn.

****

Hình 24

* + 1. Vá lỗ hổng và kiểm tra lại

Chuyển đổi sang mô hình "Pull-over-Push" (Kéo thay vì Đẩy).

* Thay vì gửi tiền ngay lập tức, hợp đồng chỉ ghi nhận số dư vào một biến mapping.
* Tạo thêm hàm withdraw() để người dùng tự chủ động rút tiền.

****

Hình 25

* Nếu kẻ tấn công từ chối nhận tiền, chỉ riêng giao dịch rút tiền của hắn bị lỗi, không ảnh hưởng đến hoạt động chung của trò chơi. Người khác vẫn có thể thay thế kẻ tấn công làm vua
  1. Lab 4: Demo GameFi (Game + Finance) - Flappy Bird Blockchain
     1. Giới thiệu kịch bản ứng dụng

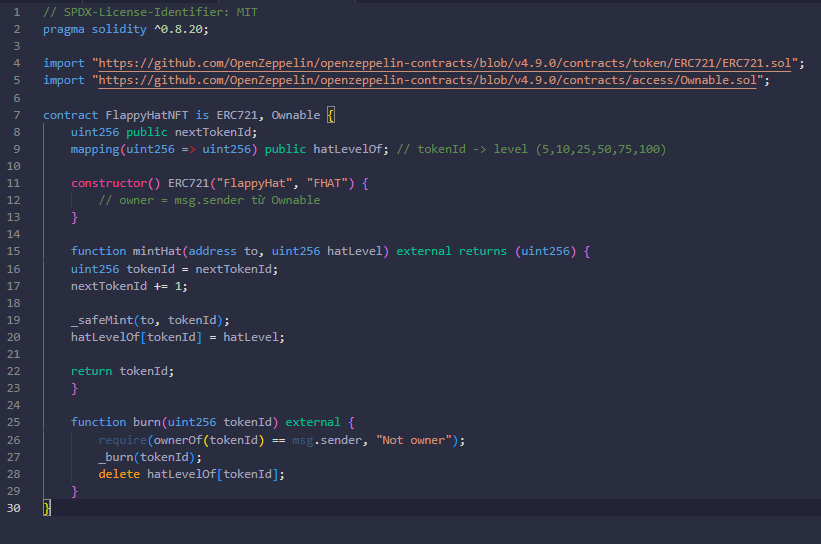
Trong bài thực nghiệm này, nhóm xây dựng một ứng dụng phi tập trung (DApp) dựa trên trò chơi kinh điển Flappy Bird. Ứng dụng tích hợp các yếu tố tài chính (GameFi) để minh họa mô hình Play-to-Earn (P2E) đơn giản.

* Mô hình hoạt động:
  + Pay-to-Play: Người chơi phải trả một khoản phí (1 ETH) để bắt đầu lượt chơi.
  + Play-to-Earn: Dựa trên điểm số đạt được, người chơi nhận phần thưởng là các vật phẩm NFT (Mũ/Hat) có cấp độ khác nhau.
  + Marketplace: Người chơi có thể bán lại các NFT này cho hệ thống để nhận lại ETH, với giá trị tăng dần theo cấp độ của mũ.
* Công nghệ sử dụng:
  + Smart Contract: Solidity (ERC721 cho NFT, logic game quản lý quỹ).
  + Frontend: HTML5 Canvas, CSS3, JavaScript.
  + Web3 Library: Ethers.js (để kết nối ví MetaMask và tương tác Blockchain).
    1. Phân tích thiết kế Hợp đồng thông minh (Smart Contract)

Hệ thống bao gồm 2 hợp đồng chính tương tác với nhau:

a) Hợp đồng   
Đây là hợp đồng tuân thủ chuẩn ERC-721, đại diện cho vật phẩm trong game (Mũ).

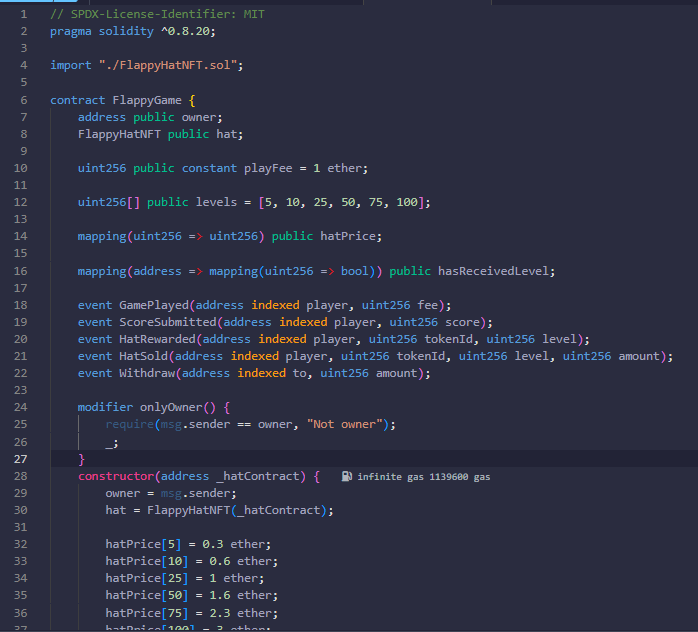
* Chức năng: Quản lý quyền sở hữu, tạo mới (mint) và hủy (burn) các NFT.
* Lưu trữ dữ liệu: Biến hatLevelOf ánh xạ từ tokenId sang cấp độ của mũ (5, 10, 25...).
* Phân quyền: Chỉ có hợp đồng Game mới có quyền gọi hàm mintHat để thưởng cho người chơi.

****

Hình 26

b) Hợp đồng   
Đây là hợp đồng trung tâm quản lý logic tài chính và tương tác trò chơi.

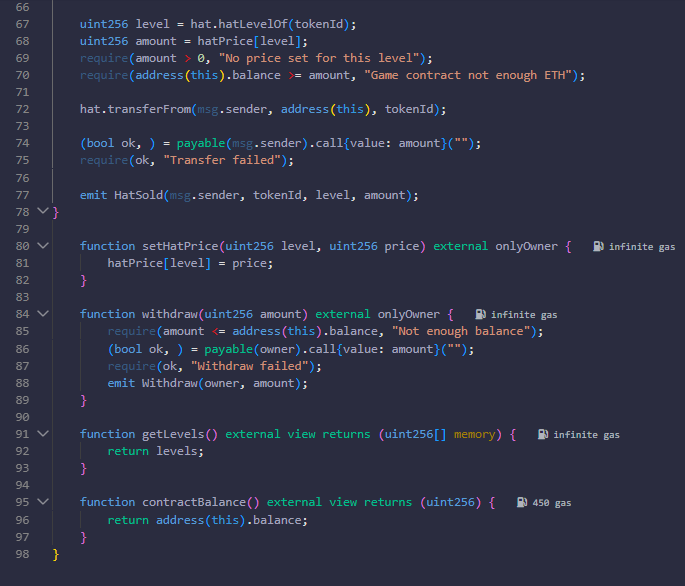
* Quản lý quỹ: Hàm playGame yêu cầu người chơi gửi đúng 1 ether. Số tiền này được giữ trong hợp đồng để trả thưởng sau này.
* Cơ chế thưởng (Reward): Hàm submitScore(uint256 score) kiểm tra điểm số. Nếu vượt qua các mốc (levels: 5, 10, 25...), hợp đồng sẽ gọi hat.mintHat để gửi NFT cho người chơi.
* Cơ chế thanh khoản (Sell): Hàm sellHat(uint256 tokenId) cho phép người chơi bán lại mũ.
  + Hệ thống xác thực quyền sở hữu.
  + Xác định giá trị mũ dựa trên level (hatPrice).
  + Chuyển NFT từ người chơi về hợp đồng (người chơi phải approve trước).
  + Chuyển ETH từ quỹ hợp đồng trả cho người chơi.

****

Hình 27

****

Hình 28

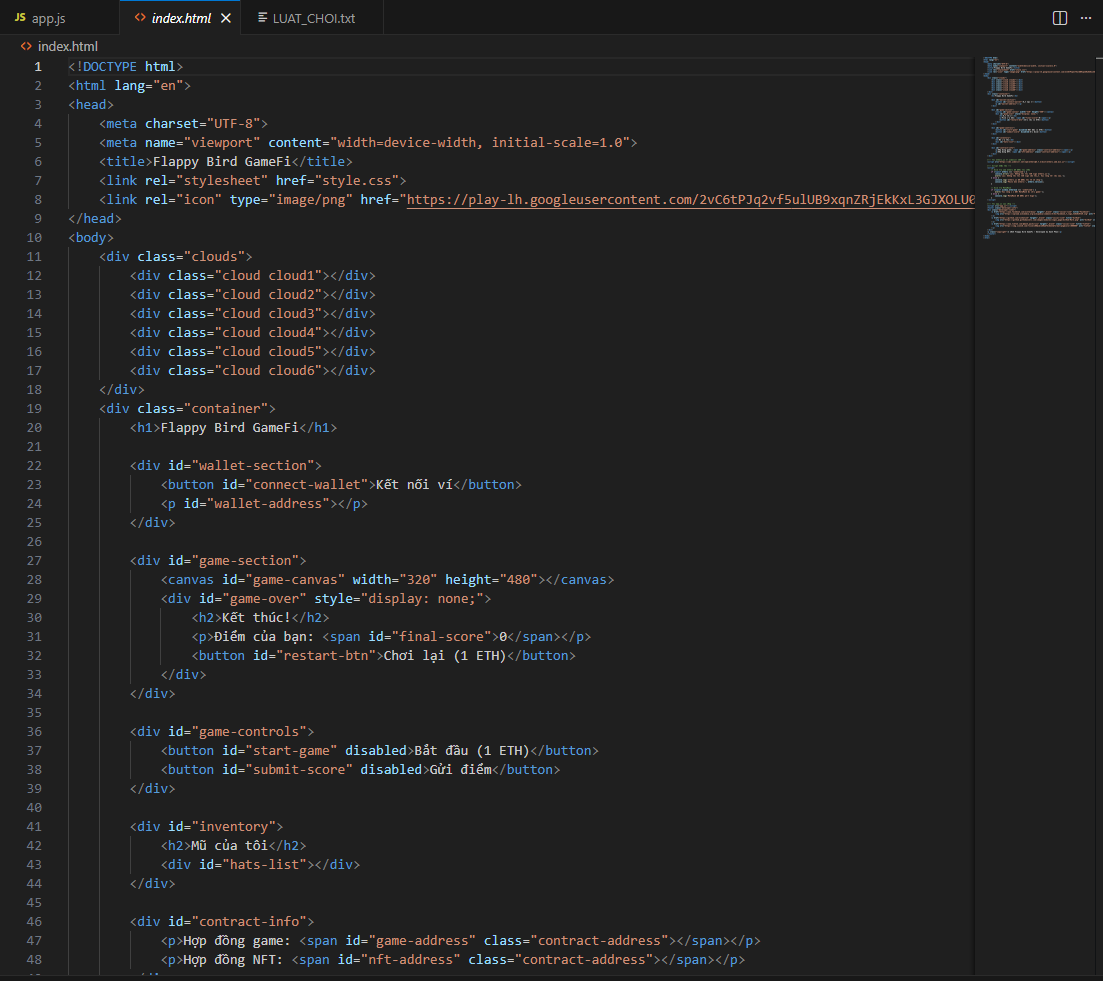
****

Hình 29

* + 1. Tích hợp Frontend và Web3

Giao diện người dùng được xây dựng bằng HTML/CSS với phong cách "Pixel Art". File app.js đóng vai trò cầu nối:

* Kết nối ví: Sử dụng window.ethereum.request để lấy địa chỉ ví người dùng.
* Xử lý giao dịch:
  + Nút "Bắt đầu": Gọi flappyGame.playGame({ value: '1.0' }).
  + Nút "Gửi điểm": Gọi flappyGame.submitScore(score).
  + Nút "Bán mũ": Gọi hat.approve() sau đó gọi flappyGame.sellHat(id).
* Đồng bộ dữ liệu: Lắng nghe các sự kiện (Events) từ Blockchain (HatRewarded, HatSold) để cập nhật giao diện Inventory (Túi đồ) ngay lập tức.

****

Hình 30

* + 1. Kịch bản kiểm thử (Demo Flow)

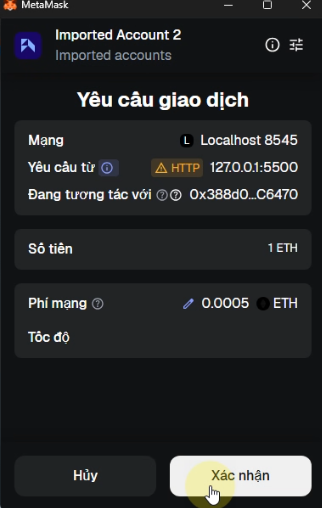
Quá trình thực nghiệm được tiến hành trên mạng Local (Ganache) với các bước sau:

Bước 1: Triển khai và Cấu hình

* Deploy FlappyHatNFT.
* Deploy FlappyGame (truyền địa chỉ NFT vào constructor).
* Quan trọng: Chuyển quyền sở hữu hoặc cấp quyền mint cho FlappyGame trong hợp đồng FlappyHatNFT (nếu có logic onlyOwner hoặc setup trong constructor).

Bước 2: Kết nối và Chơi game

* Người chơi kết nối ví MetaMask.
* Nhấn "Bắt đầu", MetaMask yêu cầu xác nhận giao dịch gửi 1 ETH.
* Người chơi điều khiển chim bay qua các ống nước. Giả sử đạt 15 điểm.

****

Hình 31

Bước 3: Nhận thưởng (Mint NFT)

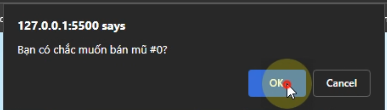
* Người chơi nhấn "Gửi điểm".
* Hợp đồng kiểm tra: 15 điểm > mức 5 và mức 10.
* Người chơi nhận được 2 NFT: Mũ Level 5 (Xám) và Mũ Level 10 (Xanh lá).
* Giao diện hiển thị danh sách mũ trong phần "Mũ của tôi".

****

Hình 32

Bước 4: Bán vật phẩm (Sell NFT)

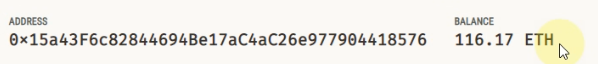
* Người chơi chọn bán Mũ Level 10.
* Hệ thống thực hiện giao dịch chuyển NFT về hợp đồng Game.
* Người chơi nhận lại 0.6 ETH (theo bảng giá hatPrice).
* Số dư ví tăng lên, số dư hợp đồng giảm đi.

****

Số dư ví trước khi bán

****

Số dư ví sau khi bán

****

* + 1. Đánh giá rủi ro bảo mật (Critical Analysis)

Mặc dù ứng dụng hoạt động đúng logic nghiệp vụ, nhưng trong quá trình xây dựng, nhóm nhận thấy một lỗ hổng bảo mật nghiêm trọng thường thấy trong các GameFi đời đầu:

* Vấn đề: Logic tính điểm (score) nằm hoàn toàn ở phía Client (JavaScript/Frontend). Hàm submitScore(uint256 score) trong Smart Contract là public và nhận tham số điểm số trực tiếp từ người dùng gửi lên.
* Nguy cơ: Một kẻ tấn công (biết lập trình) có thể không cần chơi game mà gọi trực tiếp hàm submitScore(100) thông qua Remix hoặc Etherscan. Hợp đồng sẽ tin tưởng và mint ra chiếc Mũ Level 100 (trị giá 3 ETH) ngay lập tức. Kẻ tấn công chỉ mất 1 ETH phí vào cửa (hoặc thậm chí bypass nếu logic check phí không chặt) để lấy 3 ETH.
* Giải pháp đề xuất: Để khắc phục, cần áp dụng cơ chế Oracle hoặc Backend Signing. Server sẽ xác thực quá trình chơi game, sau đó ký một chữ ký điện tử (signature) chứa điểm số. Smart Contract chỉ chấp nhận điểm số nếu đi kèm chữ ký hợp lệ của Server.
  1. Kết luận Chương 4

Chương 4 đã hiện thực hóa các lý thuyết bảo mật thông qua việc mô phỏng trực tiếp trên môi trường Blockchain cục bộ.

1. **Xác thực rủi ro:** Các thí nghiệm đã chứng minh rằng những lỗ hổng như Reentrancy, tx.origin, hay DoS không chỉ là lý thuyết mà rất dễ khai thác nếu lập trình viên chủ quan.
2. **Hiệu quả của các bản vá:** Việc áp dụng các mẫu thiết kế an toàn (Checks-Effects-Interactions, Pull-over-Push) đã ngăn chặn triệt để các cuộc tấn công trong môi trường thử nghiệm.
3. **Thách thức của GameFi:** Qua Lab 4, nhóm nhận thấy bảo mật GameFi phức tạp hơn vì nó liên quan đến cả logic trò chơi (Off-chain) và tài chính (On-chain). Việc bảo vệ tính toàn vẹn của dữ liệu đầu vào (như điểm số) là yếu tố sống còn.
4. THÁCH THỨC, TRIỂN VỌNG VÀ KẾT LUẬN
   1. Những thách thức cốt lõi
      1. Thách thức về Công nghệ: "Bộ ba bất khả thi" (The Blockchain Trilemma)

Một trong những thách thức nền tảng nhất trong kiến trúc Blockchain là "Bộ ba bất khả thi", khái niệm do Vitalik Buterin phổ biến, cho rằng một mạng lưới khó có thể tối ưu hóa đồng thời cả ba đặc tính: **Bảo mật (Security), Phi tập trung (Decentralization), và Khả năng mở rộng (Scalability)**. Các nền tảng hiện nay buộc phải chấp nhận sự đánh đổi:

* **Bitcoin và Ethereum (trước The Merge)** ưu tiên **Bảo mật** và **Phi tập trung** với hàng ngàn nút mạng trên toàn cầu. Hậu quả là **Khả năng mở rộng** bị hy sinh, dẫn đến tốc độ giao dịch chậm (khoảng 7 TPS cho Bitcoin) và chi phí cao.
* Ngược lại, các blockchain như **Solana** ưu tiên **Khả năng mở rộng**, đạt tốc độ hàng ngàn TPS. Tuy nhiên, điều này phải đánh đổi bằng việc yêu cầu phần cứng rất cao, dẫn đến số lượng node xác thực (validator) ít hơn đáng kể, làm giảm mức độ **Phi tập trung**.

Bài toán này buộc các nhà phát triển phải tìm ra sự cân bằng hoặc các giải pháp sáng tạo như Layer-2 để khắc phục, vốn sẽ được đề cập ở phần sau.

* + 1. . Thách thức về Con người và Quản trị: Bài học từ sự sụp đổ của FTX

Công nghệ dù an toàn đến đâu cũng không thể loại bỏ rủi ro từ con người. Vụ sụp đổ của sàn giao dịch FTX năm 2022 là một minh chứng đanh thép. Về mặt kỹ thuật, blockchain vẫn hoạt động minh bạch, nhưng hoạt động quản trị **ngoài chuỗi (off-chain)** của FTX đã thất bại hoàn toàn.

Các báo cáo cho thấy FTX đã tự ý sử dụng tiền gửi của khách hàng cho các khoản đầu tư rủi ro cao thông qua công ty con Alameda Research. Quyền lực tập trung vào tay một nhóm cá nhân quá nhỏ, thiếu sự kiểm soát nội bộ và minh bạch tài chính. Khi thị trường biến động và người dùng rút tiền ồ ạt, sự thiếu hụt thanh khoản đã làm sụp đổ cả một đế chế.

Bài học FTX nhấn mạnh rằng: an ninh của một hệ sinh thái không chỉ nằm ở mã nguồn. Sự yếu kém trong quản trị, thiếu quy định pháp lý và lòng tham của con người có thể phá hủy niềm tin và giá trị nhanh hơn bất kỳ cuộc tấn công mạng nào.

* 1. Triển vọng phát triển trong tương lai
     1. Giải pháp mở rộng Layer-2: Hướng đi tất yếu

Để giải quyết "Bộ ba bất khả thi", các giải pháp Layer-2 đã trở thành xu hướng chủ đạo. Đây là các giao thức được xây dựng "bên trên" blockchain chính (Layer-1), giúp xử lý giao dịch ngoài chuỗi một cách nhanh chóng và rẻ hơn, sau đó chỉ gửi kết quả cuối cùng về Layer-1 để xác thực. Các giải pháp phổ biến bao gồm:

* **Rollups (Optimistic & ZK-Rollups):** Nén hàng trăm giao dịch thành một bản ghi duy nhất trên Layer-1, giúp giảm đáng kể chi phí và tăng tốc độ xử lý lên hàng ngàn TPS.
* **Sidechains:** Các blockchain độc lập chạy song song và kết nối với chuỗi chính thông qua các "cầu nối" (bridge), ví dụ như Polygon.

Sự trưởng thành của Layer-2 đang biến các ứng dụng Web3 từ những thử nghiệm đắt đỏ trở nên khả thi cho hàng triệu người dùng, mở đường cho việc áp dụng rộng rãi.

* + 1. Tương lai của Web3 và Metaverse

Web3 và Metaverse được xem là chương tiếp theo của Internet, với Blockchain đóng vai trò xương sống.

* **Web3** là tầm nhìn về một Internet phi tập trung, nơi người dùng thực sự sở hữu dữ liệu và tài sản kỹ thuật số của mình thông qua các ví mã hóa, thay vì bị kiểm soát bởi các tập đoàn công nghệ lớn.
* **Metaverse** là không gian ảo 3D nơi người dùng có thể tương tác, làm việc và giải trí. Trong đó, Blockchain cung cấp nền tảng kinh tế thông qua NFT (đất đai, vật phẩm ảo) và tiền mã hóa, tạo ra một nền kinh tế kỹ thuật số thực thụ.

Sự kết hợp giữa Web3 (cơ sở hạ tầng sở hữu) và Metaverse (không gian trải nghiệm) hứa hẹn sẽ tái định hình cách chúng ta tương tác xã hội, giải trí và thương mại trong thập kỷ tới.

* 1. Kết luận toàn diện

Hành trình của bài báo cáo đã đi từ việc hệ thống hóa **lý thuyết** nền tảng, khám phá các **ứng dụng** thực tiễn trị giá hàng tỷ đô la, và đi đến trọng tâm là **thực nghiệm an ninh** Hợp đồng thông minh. Quá trình mô phỏng tấn công và phòng thủ 3 lỗ hổng phổ biến đã cho thấy một sự thật rõ ràng: một sai sót nhỏ trong mã lệnh có thể bị khai thác để gây ra thiệt hại tài chính khổng lồ. Con số **2.2 tỷ USD** bị đánh cắp trong năm 2024 theo báo cáo của Chainalysis chính là minh chứng cho mức độ nghiêm trọng của vấn đề này.

Bài học cốt lõi rút ra từ đề tài là **an ninh không phải là một tính năng, mà là một yêu cầu bắt buộc**. Việc áp dụng các mẫu thiết kế an toàn như "Checks-Effects-Interactions" để chống tấn công Tái nhập, hay "Pull-over-Push" để tránh Tấn công Từ chối Dịch vụ không chỉ là lý thuyết suông mà là những biện pháp sống còn. Thực nghiệm trên môi trường Ganache đã chứng minh hiệu quả của các kỹ thuật phòng thủ này, khẳng định tầm quan trọng của quy trình kiểm thử (audit) và đánh giá rủi ro trước khi triển khai bất kỳ ứng dụng nào.

Cuối cùng, Blockchain là một công nghệ mang tính cách mạng với tiềm năng to lớn, nhưng nó không phải là một viên đạn bạc. Để công nghệ này phát triển bền vững và chiếm được lòng tin của công chúng, việc hiểu biết sâu sắc và ưu tiên hàng đầu cho an ninh là yếu tố then chốt. Chính sự cẩn trọng trong từng dòng mã lệnh mới là nền tảng vững chắc nhất để xây dựng một tương lai phi tập trung an toàn và đáng tin cậy.

Tài liệu tham khảo:

https://www.geeksforgeeks.org/ethical-hacking/types-of-blockchain/

https://www.blockchain-council.org/blockchain/types-of-blockchains-explained-public-vs-private-vs-consortium/

https://www.geeksforgeeks.org/dbms/difference-between-blockchain-and-a-database/

https://vinachg.vn/giai-phap-phan-mem-chong-hang-gia-tren-nen-tang-so/?nowprocket=1

https://www.christies.com/en/stories/monumental-collage-by-beeple-is-first-purely-digital-artwork-nft-to-come-to-auction-0463a2c0f3174b17997fba8a1fe4c865

https://dappradar.com/dapp/axie-infinity?range-ha=1y&range-mct=all

https://cryptoslate.com/ethereum-name-service-saw-2-2m-ens-names-registered-in-2022/

https://e-estonia.com/story/

https://www.gate.com/vi/post/status/13486410

https://101blockchains.com/reentrancy-attack/

https://docs.soliditylang.org/en/latest/security-considerations.html

https://thecryptorecruiters.io/layer-2-scaling-solutions/

https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos#prevention

https://www.moonpay.com/learn/blockchain/what-is-the-blockchain-trilemma

https://www.binance.com/vi/square/post/21737927642225

https://www.researchgate.net/figure/The-Scalability-Trilemma\_fig1\_342639281

https://walton.uark.edu/insights/posts/culture-controls-and-corporate-governance-lessons-from-the-ftx-fiasco.php

https://simon.rochester.edu/blog/deans-corner/5-lessons-learn-collapse-ftx

https://thecryptorecruiters.io/layer-2-scaling-solutions/

https://www.brokerhivex.com/en/investment-academy/68c67bf24a949

https://metana.io/blog/top-15-web3-trends-to-watch-in-2025/

https://www.chainalysis.com/blog/crypto-hacking-stolen-funds-2025/

https://coinlaw.io/smart-contract-security-risks-and-audits-statistics/