Bộ Giáo Dục Và Đào Tạo

Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh

**Khoa Công Nghệ Thông Tin**



**MÔN HỌC : MẠNG KHÔNG DÂY**

**ĐỀ TÀI : TẤN CÔNG VÀ BẢO VỆ GIAO THỨC ZIGBEE TRÊN THIẾT BỊ IOT**

**Giảng Viên Hướng Dẫn** : ThS. CAO TIẾN THÀNH

**Nhóm 11**

**Thành Viên :**

1. Phạm Quốc Huy – MSSV: 22DH111305
2. Võ Văn Thần Thái – MSSV: 22DH114737
3. Võ Công Hiếu – MSSV: 22DH111085

*Tp. Hồ chí minh, Ngày 8 tháng 4 năm 2025*

**MỤC LỤC**

[Danh mục hình ảnh 4](#_Toc194997717)

[Danh mục bảng 5](#_Toc194997718)

[Lời giới thiệu 6](#_Toc194997719)

[Lời cảm ơn 7](#_Toc194997720)

[Nhận xét của giảng viên 8](#_Toc194997721)

[CHƯƠNG 1: GIỚI THIỆU 9](#_Toc194997722)

[1. Giới thiệu về đề tài 9](#_Toc194997723)

[1.1 Đặt vấn đề 9](#_Toc194997724)

[2. Mục tiêu nghiên cứu đề tài 9](#_Toc194997725)

[2.1. Mục tiêu nghiên cứu 9](#_Toc194997726)

[2.2 Công cụ sử dụng 9](#_Toc194997727)

[2.3 Phương pháp nghiên cứu 10](#_Toc194997728)

[2.4 Kết quả mong đợi 10](#_Toc194997729)

[3. Phạm vi nghiên cứu 10](#_Toc194997730)

[3.1 Tổng quan về Zigbee 10](#_Toc194997731)

[3.2 Các hình thức tấn công Zigbee 11](#_Toc194997732)

[3.3 Các biện pháp bảo vệ giao thức Zigbee 11](#_Toc194997733)

[3.4 Thực nghiệm và đánh giá 11](#_Toc194997734)

[CHƯƠNG 2: CƠ SỞ LÍ THUYẾT 12](#_Toc194997735)

[1. Giao thức Zigbee LÀ GÌ 12](#_Toc194997736)

[2. Nguyên lý hoạt động giao thức Zigbee ? 13](#_Toc194997737)

[3. Thiết bị IOT là gì ? 15](#_Toc194997738)

[4. Tầm quan trọng của Zigbee 16](#_Toc194997739)

[5. Ưu nhược điểm khi dùng thiết bị thông minh Zigbee 17](#_Toc194997740)

[6. Ứng dụng của Zigbee đối với các thiết bị IoT 18](#_Toc194997741)

[6.1 Nhà thông minh (Smart Home) 18](#_Toc194997742)

[6.2 Thành phố thông minh (Smart City) 18](#_Toc194997743)

[6.3 Y tế thông minh (Smart Healthcare) 19](#_Toc194997744)

[7. SO SÁNH Zigbee với các công nghệ mạng không dây khác 20](#_Toc194997745)

[8. Công cụ KillerBee 21](#_Toc194997746)

[9. Công cụ Zigator 22](#_Toc194997747)

[10. Công cụ Scapy 23](#_Toc194997748)

[11. So sánh các công cụ KillerBee, Zigator và Scapy 24](#_Toc194997749)

[CHƯƠNG 3 : PHƯƠNG PHÁP THỰC HIỆN 25](#_Toc194997750)

[1. Tổng quan hệ thống 25](#_Toc194997751)

[2. Yêu cầu hệ thống 25](#_Toc194997752)

[a. Cấu hình máy chủ Ubuntu 25](#_Toc194997753)

[b. Phần mềm cần thiết 25](#_Toc194997754)

[3. Kiến trúc hệ thống 25](#_Toc194997755)

[a) Hệ thống bao gồm các thành phần chính: 25](#_Toc194997756)

[b) ZigBee Emulator: 27](#_Toc194997757)

[c) Quy trình hoạt động: 27](#_Toc194997758)

[d) Kết nối với máy chủ: 27](#_Toc194997759)

[e) Hiệu năng đồ họa: 27](#_Toc194997760)

[CHƯƠNG 4 : TRIỂN KHAI 28](#_Toc194997761)

[1. KillerBee 28](#_Toc194997762)

[Bước 1: Cài đặt các gói cần thiết 28](#_Toc194997763)

[Bước 2: Cài đặt thư viện phụ thuộc 29](#_Toc194997764)

[Bước 3: Tải KillerBee từ GitHub 30](#_Toc194997765)

[Bước 4: Cài đặt KillerBee 30](#_Toc194997766)

[Bước 5: Kiểm tra KillerBee đã cài đặt thành công 31](#_Toc194997767)

[2. Zigator 32](#_Toc194997768)

[3. Scapy 33](#_Toc194997769)

[Bước 1: Clone repository Scapy từ GitHub 33](#_Toc194997770)

[Bước 2: Di chuyển vào thư mục Scapy 33](#_Toc194997771)

[Bước 3: Cài đặt Scapy bằng setup.py 34](#_Toc194997772)

[Bước 4: Kiểm tra cài đặt Scapy 34](#_Toc194997773)

[4.Demo bằng công cụ Scapy 35](#_Toc194997774)

[Bước 1: Kiểm tra giao diện mạng Loopback 35](#_Toc194997775)

[Bước 2: Sniffing Attack - Bắt gói tin Zigbee 35](#_Toc194997776)

[Bước 3: Gửi gói tin Zigbee giả lập 36](#_Toc194997777)

[Bước 4: Lưu gói tin đã bắt được 36](#_Toc194997778)

[Bước 5: Phát lại gói tin - Replay Attack 36](#_Toc194997779)

[Bước 6: Kiểm tra xem Replay Attack có thành công không 37](#_Toc194997780)

[5. Demo bảo mật hệ thống zigbee 37](#_Toc194997781)

[5.1. Giới thiệu về Demo 37](#_Toc194997782)

[5.2. Cấu trúc và Thành phần của Demo 37](#_Toc194997783)

[5.2.1 Lớp ZigBeePacket 37](#_Toc194997784)

[5.2.2 Lớp ZigBeeNetwork 37](#_Toc194997785)

[5.2.3 Lớp ZigBeeAttacker 38](#_Toc194997786)

[5.2.4. Lớp ZigBeeSecurityAdvisor 38](#_Toc194997787)

[5.3 Kỹ Thuật Mã Hóa Trong Demo 38](#_Toc194997788)

[5.3.1 Mã hóa AES-128 (CTR Mode) 38](#_Toc194997789)

[5.3.2 Quá trình giải mã 38](#_Toc194997790)

[5.4 Các Biện Pháp Bảo Mật Được Triển Khai 38](#_Toc194997791)

[5.4.1 Mã hóa dữ liệu 38](#_Toc194997792)

[5.4.2 Quản lý khóa 39](#_Toc194997793)

[5.4.3 Các khuyến nghị bảo mật nâng cao 39](#_Toc194997794)

[CHƯƠNG 5 : ĐÁNH GIÁ KẾT LUẬN 40](#_Toc194997795)

[1.Đánh giá kết quả nghiên cứu 40](#_Toc194997796)

[1.1Xác định lỗ hổng Zigbee: 40](#_Toc194997797)

[1.2.Thực hiện các cuộc tấn công: 40](#_Toc194997798)

[1.3.Triển khai các biện pháp bảo vệ: 41](#_Toc194997799)

[2. Hạn chế và hướng phát triển 42](#_Toc194997800)

[3. Kết luận 42](#_Toc194997801)

[4.Tài liệu tham khảo 43](#_Toc194997802)

# Danh mục hình ảnh

[Hình 1. Giao thức Zigbee 14](#_Toc193741605)

[Hình 2. Nguyên lý hoạt động của Zigbee 15](#_Toc193741606)

[Hình 3. Thiết bị IoT 16](#_Toc193741607)

[Hình 4. Nhà thông minh sử dụng giao thức Zigbee với các thiết bị Iot 19](#_Toc193741608)

[Hình 5. Thành phố thông minh sử dụng giao thức Zigbee 20](#_Toc193741609)

[Hình 6. Y tế thông minh 21](#_Toc193741610)

[Hình 7. Sơ đồ mạng 27](#_Toc193741611)

[Hình 8. Cập nhật hệ thống 29](#_Toc193741612)

[Hình 9. Công cụ quản lí thư viện python3 30](#_Toc193741613)

[Hình 10. Cài đặt các thư viện phụ thuộc 31](#_Toc193741614)

[Hình 11. Clone KillerBee 31](#_Toc193741615)

[Hình 12. Cài đặt KillerBee 32](#_Toc193741616)

[Hình 13. Kiểm tra KillerBee cài đặt thành công 33](#_Toc193741617)

[Hình 14. Tạo môi trường ảo hóa mới 33](#_Toc193741618)

[Hình 15. Cài đặt Zigator 34](#_Toc193741619)

[Hình 16. Kiểm tra cài đặt Zigator 34](#_Toc193741620)

[Hình 17. Cài đặt Scapy 35](#_Toc193741621)

[Hình 18. Kiểm tra cài đặt Scapy 36](#_Toc193741622)

[Hình 19. Kết quả Sniffing 41](#_Toc193741623)

[Hình 20. kết quả Replay Attack 42](#_Toc193741624)

[Hình 21. kết quả bảo mật 43](#_Toc193741625)

# Danh mục bảng

[Bảng 1. So sánh các công nghệ mạng không dây 20](#_Toc192611146)

[Bảng 2. So sánh các công cụ KillerBee, Zigator và Scapy 24](#_Toc192611147)

# Lời giới thiệu

Kính thưa thầy cô nhóm chúng em xin được trình bày về đề tài Tấn công và bảo vệ giao thức Zigbee trên thiết bị IoT. Trong bối cảnh Internet of Things (IoT) phát triển mạnh mẽ, giao thức Zigbee được sử dụng rộng rãi trong các hệ thống nhà thông minh, tự động hóa công nghiệp và y tế nhờ khả năng tiết kiệm năng lượng, kết nối linh hoạt và chi phí thấp. Tuy nhiên, do đặc thù truyền thông không dây và cơ chế bảo mật còn hạn chế, Zigbee trở thành mục tiêu của nhiều cuộc tấn công mạng, gây ra nguy cơ mất an toàn cho thiết bị IoT.

Zigbee là một giao thức truyền thông không dây dựa trên tiêu chuẩn IEEE 802.15.4, chuyên dùng cho các thiết bị IoT, nhà thông minh, hệ thống cảm biến và tự động hóa công nghiệp. Với ưu điểm tiêu thụ điện năng thấp, cấu trúc mạng linh hoạt (Mesh, Star, Tree) và khả năng mở rộng cao, Zigbee đang trở thành một giải pháp phổ biến trong các ứng dụng cảm biến không dây, giám sát môi trường, và điều khiển thiết bị từ xa.

Tuy nhiên, do được thiết kế chủ yếu để tiết kiệm năng lượng và chi phí, mạng Zigbee cũng tiềm ẩn nhiều rủi ro bảo mật, đặc biệt là trong các hệ thống không được cấu hình đúng cách. Các cuộc tấn công có thể nhắm vào việc giả mạo thiết bị, nghe lén dữ liệu, tấn công từ chối dịch vụ (DoS) hoặc tấn công Man-in-the-Middle (MITM) để chiếm quyền điều khiển hệ thống.

KillerBee là một bộ công cụ chuyên dụng để kiểm thử bảo mật mạng Zigbee và IEEE 802.15.4. Nó được phát triển nhằm giúp các chuyên gia bảo mật đánh giá mức độ an toàn của các thiết bị IoT và hệ thống Zigbee.Với tốc độ phát triển nhanh của IoT, việc đảm bảo an toàn cho các thiết bị sử dụng Zigbee là cực kỳ quan trọng. Đề tài này không chỉ giúp phát hiện lỗ hổng mà còn góp phần nâng cao an ninh trong các hệ thống không dây hiện nay.

# Lời cảm ơn

Em xin gửi lời cảm ơn chân thành đến thầy Thành, người đã tận tình hướng dẫn, góp ý và hỗ trợ em trong suốt quá trình nghiên cứu và hoàn thành đề tài này. Em xin cảm ơn trường đại học ngoại ngữ và tin học đã cung cấp tài liệu, cơ sở vật chất và môi trường học tập để em thực hiện đề tài này. Dù đã cố gắng hoàn thiện, nhưng chắc chắn vẫn còn thiếu sót. Em mong nhận được sự góp ý để đề tài được hoàn thiện hơn. Em xin chân thành cảm ơn!

# Nhận xét của giảng viên

# CHƯƠNG 1: GIỚI THIỆU

## 1. Giới thiệu về đề tài

Tấn công và bảo vệ giao thức Zigbee trên thiết bị IoT

### 1.1 Đặt vấn đề

Trong bối cảnh Internet of Things (IoT) phát triển mạnh mẽ, giao thức Zigbee trở thành một lựa chọn phổ biến trong các hệ thống nhà thông minh, điều khiển công nghiệp và cảm biến không dây. Với khả năng tiết kiệm năng lượng, hỗ trợ mạng mesh và chi phí thấp, Zigbee được sử dụng rộng rãi trong nhiều thiết bị IoT.

Tuy nhiên, các lỗ hổng bảo mật của Zigbee đã bị khai thác, đặc biệt là sniffing attack (nghe lén) và replay attack (tấn công phát lại), có thể dẫn đến xâm nhập trái phép, điều khiển thiết bị từ xa, đánh cắp dữ liệu.

Vì vậy, việc nghiên cứu các phương pháp tấn công và bảo vệ giao thức Zigbee là rất quan trọng để đảm bảo tính an toàn và bảo mật cho các hệ thống IoT.

## 2. Mục tiêu nghiên cứu đề tài

Mục tiêu nghiên cứu đề tài: Tấn công và bảo vệ giao thức Zigbee trên thiết bị IoT

### 2.1. Mục tiêu nghiên cứu

* Phân tích giao thức Zigbee: Hiểu rõ cơ chế hoạt động, kiến trúc mạng, và các lỗ hổng bảo mật tiềm ẩn.
* Thực hiện các cuộc tấn công: Áp dụng các kỹ thuật như sniffing, replay attack, và các phương pháp khai thác khác để đánh giá tính bảo mật của giao thức Zigbee.
* Đề xuất giải pháp bảo mật: Xây dựng các phương pháp như mã hóa dữ liệu, xác thực thiết bị, và cải tiến cơ chế bảo mật nhằm giảm thiểu nguy cơ tấn công.
* Mô phỏng hệ thống IoT: Xây dựng mô hình thực tế sử dụng Zigbee, thử nghiệm các cuộc tấn công và kiểm chứng giải pháp bảo mật đề xuất.

### 2.2 Công cụ sử dụng

**KillerBee**

* Bộ công cụ tấn công giao thức Zigbee mã nguồn mở.
* Hỗ trợ nhiều kiểu tấn công như sniffing, replay attack, packet injection.
* Tương thích với phần cứng như Atmel RZUSBstick, TI CC2531, ApiMote.

### 2.3 Phương pháp nghiên cứu

Phân tích giao thức Zigbee:

* Tìm hiểu kiến trúc mạng (Coordinator, Router, End Device).
* Phân tích lớp vật lý, lớp MAC, và lớp ứng dụng của Zigbee.
* Xác định các điểm yếu trong cơ chế bảo mật.

Thực hiện tấn công:

* Sniffing: Dùng KillerBee để chặn gói tin Zigbee.
* Replay Attack: Gửi lại các gói tin đã bắt được để điều khiển thiết bị.
* Traffic Analysis: Phân tích luồng dữ liệu để phát hiện thông tin quan trọng.

Đề xuất giải pháp bảo mật:

* Mã hóa AES để bảo vệ dữ liệu.
* Cơ chế xác thực thiết bị để ngăn chặn tấn công giả mạo.
* Phát hiện xâm nhập (IDS) dựa trên phân tích hành vi bất thường.

Mô phỏng và kiểm thử:

* Xây dựng hệ thống IoT sử dụng Zigbee.
* Thực hiện các cuộc tấn công thử nghiệm.
* Triển khai giải pháp bảo mật và đánh giá hiệu quả.

### 2.4 Kết quả mong đợi

* Xác định các lỗ hổng bảo mật quan trọng trong Zigbee.
* Chứng minh khả năng thực hiện các cuộc tấn công trên thiết bị thực tế.
* Đưa ra giải pháp bảo mật khả thi giúp nâng cao an toàn cho các thiết bị IoT sử dụng Zigbee.
* Tạo tài liệu và công cụ hướng dẫn để các nhà nghiên cứu và kỹ sư bảo mật có thể áp dụng.

## 3. Phạm vi nghiên cứu

Đề tài tập trung vào việc phân tích các phương thức tấn công và biện pháp bảo vệ giao thức Zigbee trên thiết bị IoT. Phạm vi nghiên cứu bao gồm:

### 3.1 Tổng quan về Zigbee

* Cấu trúc mạng và nguyên lý hoạt động.
* Ứng dụng trong IoT, đặc biệt là nhà thông minh và công nghiệp.

### 3.2 Các hình thức tấn công Zigbee

* Nghe lén và đánh cắp dữ liệu: Kẻ tấn công có thể sử dụng công cụ như Zigator hoặc KillerBee để bắt gói tin Zigbee.
* Tấn công phát lại (Replay Attack): Gửi lại gói tin cũ để giả mạo lệnh điều khiển thiết bị.
* Tấn công Man-in-the-Middle (MITM): Chèn gói tin giả mạo, chiếm quyền kiểm soát thiết bị.
* Tấn công từ chối dịch vụ (DoS): Gây nhiễu hoặc gửi quá nhiều gói tin để làm gián đoạn hoạt động của hệ thống.
* Tấn công vào cơ chế mã hóa: Khai thác lỗ hổng trong quản lý khóa để phá vỡ bảo mật AES-128.

### 3.3 Các biện pháp bảo vệ giao thức Zigbee

* Cải thiện bảo mật truyền thông: Áp dụng quản lý khóa an toàn, xác thực thiết bị.
* Giám sát và phát hiện tấn công: Sử dụng hệ thống phát hiện xâm nhập (IDS) để phát hiện các hành vi bất thường.
* Bảo vệ phần cứng thiết bị: Ngăn chặn truy cập vật lý để tránh lộ khóa mã hóa.

### 3.4 Thực nghiệm và đánh giá

* Thử nghiệm tấn công trên thiết bị Zigbee thực tế hoặc mô phỏng bằng công cụ chuyên dụng.
* Đánh giá hiệu quả của các biện pháp bảo vệ và đề xuất giải pháp cải thiện.

# CHƯƠNG 2: CƠ SỞ LÍ THUYẾT

1. **Giao thức Zigbee LÀ GÌ**

Mạng Zigbee là gì? Mạng Zigbee là một công nghệ không dây đã được ra đời cách đây hơn một thập kỷ. Nó được xây dựng dựa trên tiêu chuẩn 802.15.4 của IEEE. Là một giải pháp thay thế cho mạng wifi và bluetooth trên một số ứng dụng. Trong đó sẽ bao gồm các thiết bị sử dụng ít năng lượng. Chẳng hạn như bóng [đèn thông minh](https://fptsmarthome.vn/thiet-bi/den-thong-minh" \t "_blank) và [công tắc thông minh](https://fptsmarthome.vn/thiet-bi/cong-tac-thong-minh" \t "_blank) có thể kết nối thông qua mạng Zigbee nếu như chúng được thiết kế sử dụng cùng 1 chuẩn kết nối.

Điều đặc biệt của Zigbee là nó sử dụng mức tiêu thụ điện năng thấp, cho phép nó hoạt động trong thời gian dài trên pin. Nó cũng có khả năng kết nối hơn một trăm thiết bị trên một mạng lưới đơn, điều này làm cho nó trở thành một lựa chọn tuyệt vời cho các ứng dụng nhà thông minh và IoT.

Ngoài ra, Zigbee còn cung cấp tính năng bảo mật cao, cho phép các thiết bị trao đổi thông tin một cách an toàn và bảo mật. Nó cũng hỗ trợ nhiều ứng dụng khác nhau, bao gồm đo lường nhiệt độ, đo lường độ ẩm, điều khiển ánh sáng và điều khiển năng lượng.

Zigbee được sử dụng rộng rãi trong các ứng dụng nhà thông minh, từ điều khiển ánh sáng và điều khiển nhiệt độ đến các hệ thống an ninh và giám sát. Nó cũng được sử dụng trong các ứng dụng IoT như theo dõi thời tiết và các hệ thống đo lường khoảng cách.

Tóm lại, Zigbee là một công nghệ mạng không dây tiên tiến và tiết kiệm năng lượng cho phép các thiết bị thông minh giao tiếp với nhau một cách an toàn và bảo mật. Nó được sử dụng rộng rãi trong các ứng dụng nhà thông minh và IoT, và được coi là một lựa chọn tuyệt vời cho các hệ thống kết nối thiết bị thông minh.



**Hình 1. Giao thức Zigbee**

1. **Nguyên lý hoạt động giao thức Zigbee ?**

Zigbee hoạt động dựa trên chuẩn IEEE 802.15.4, sử dụng mô hình mạng không dây tiêu thụ năng lượng thấp để truyền dữ liệu giữa các thiết bị. Nó có thể hoạt động theo ba mô hình mạng chính: mạng sao (Star), mạng lưới (Mesh) và mạng cây (Tree). Trong đó, mạng Mesh phổ biến nhất nhờ khả năng mở rộng và tự phục hồi khi một nút bị lỗi.

Mạng Zigbee gồm ba loại thiết bị chính: Zigbee Coordinator (Điều phối viên - ZC), Zigbee Router (Bộ định tuyến - ZR) và Zigbee End Device (Thiết bị đầu cuối - ZED). Điều phối viên là thiết bị trung tâm duy nhất có quyền thiết lập mạng, quản lý địa chỉ và bảo mật dữ liệu. Router giúp mở rộng phạm vi mạng bằng cách chuyển tiếp dữ liệu giữa các thiết bị, trong khi End Device chỉ có thể giao tiếp với Router hoặc Coordinator mà không thể chuyển tiếp dữ liệu, giúp tiết kiệm năng lượng.

Cơ chế truyền dữ liệu

Zigbee hỗ trợ ba phương thức truyền dữ liệu: Unicast (truyền đến một thiết bị cụ thể), Broadcast (truyền đến tất cả thiết bị trong mạng) và Multicast (truyền đến một nhóm thiết bị xác định). Khi một thiết bị gửi dữ liệu, nó sử dụng thuật toán AODV (Ad hoc On-Demand Distance Vector) để tìm đường đi tối ưu đến thiết bị đích. Nếu thiết bị đích nằm ngoài phạm vi truyền trực tiếp, gói tin sẽ được chuyển tiếp qua nhiều Router cho đến khi đến đúng đích.

Zigbee sử dụng cơ chế CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) để tránh xung đột khi nhiều thiết bị truyền dữ liệu đồng thời. Một thiết bị trước khi truyền dữ liệu sẽ kiểm tra kênh truyền có đang bận hay không. Nếu kênh trống, nó sẽ gửi dữ liệu ngay lập tức; nếu kênh đang bận, thiết bị sẽ đợi một khoảng thời gian ngẫu nhiên trước khi thử lại.

Tối ưu hóa năng lượng

Một trong những ưu điểm quan trọng của Zigbee là khả năng tiết kiệm năng lượng. Các thiết bị End Device có thể chuyển sang chế độ ngủ (sleep mode) khi không hoạt động để giảm mức tiêu thụ pin. Điều này giúp các thiết bị IoT chạy bằng pin có thể hoạt động trong nhiều năm mà không cần thay pin thường xuyên. Khi cần truyền dữ liệu, End Device sẽ thức dậy, gửi dữ liệu và sau đó quay lại chế độ ngủ.



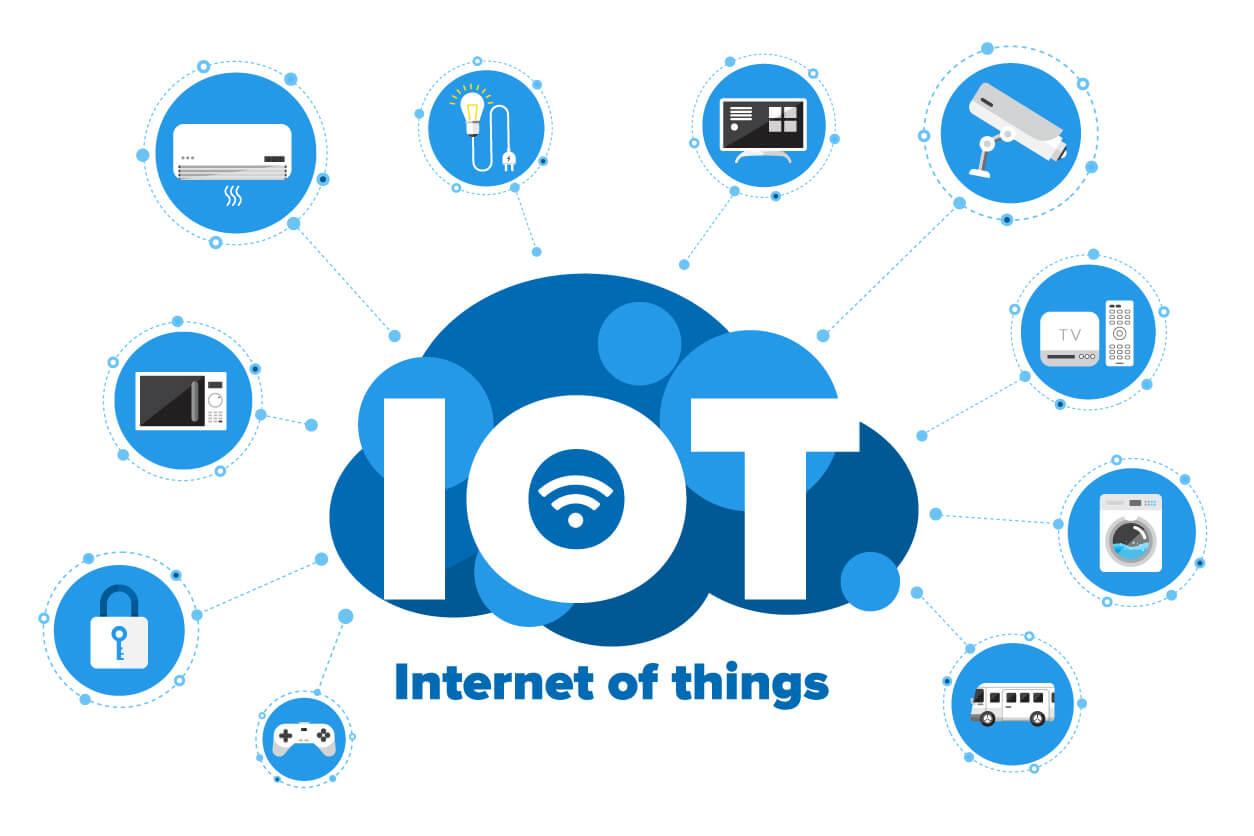
**Hình 2. Nguyên lý hoạt động của Zigbee**

1. **Thiết bị IOT là gì ?**

[Thiết bị IoT](https://e-vnpt.vn/thiet-bi-iot-la-gi-ai-se-dung-thiet-bi-iot/) (Internet of Things) là thiết bị kết nối internet và có khả năng thu thập, xử lý và chia sẻ dữ liệu một cách tự động. Thiết bị này được trang bị các cảm biến và phần mềm để giao tiếp với nhau hoặc với mạng internet. Các thiết bị IoT thông thường được sử dụng để tạo ra các hệ thống tự động hoặc để giám sát các quy trình hoạt động trong nhà, công ty hay ngành sản xuất.

Các thiết bị IoT cũng được sử dụng để tạo ra các hệ thống thông minh, giúp tăng tính an toàn và tiết kiệm chi phí. Ví dụ, các hệ thống nhà thông minh có thể được lập trình để tự động mở và đóng cửa, điều khiển đèn, điều hòa, máy lạnh… theo thời gian và điều kiện môi trường. Điều này giúp tiết kiệm năng lượng và tạo ra sự tiện ích cho người sử dụng.

Tuy nhiên, việc sử dụng các thiết bị IoT cũng mang lại một số rủi ro về an ninh mạng và bảo mật thông tin. Vì vậy, quản lý bảo mật và giám sát kỹ thuật của các thiết bị IoT là rất quan trọng để đảm bảo an toàn cho người sử dụng và ngăn chặn các cuộc tấn công từ phía hacker.



**Hình 3. Thiết bị IoT**

1. **Tầm quan trọng của Zigbee**

Công nghệ Zigbee có các nút hoạt động như một mạng lưới. Ngay cả khi có nút nào đó phát hiện tín hiệu bị lỗi, các thiết bị vẫn được hoạt động một cách ổn định. Để đánh giá cụ thể hơn về tầm quan trọng của mạng Zigbee, chúng ta sẽ đi qua từng yếu tố sau đây:

**Tính bảo mật cao**

Để đánh giá về vấn đề bảo mật của mạng Zigbee, loại mạng này được bảo mật với các khóa đối xứng 128 bit. Cách mã hóa này đạt tiêu chuẩn cao được sử dụng trong các ứng dụng trực tuyến của ngân hàng hay cả trong các cơ quan chính phủ. Chính vì thế khi sử dụng Zigbee sẽ làm tăng khả năng bảo mật. Người dùng cũng yên tâm hơn khi sử dụng hệ thống nhà smartphone có kết nối Zigbee.

**Mạng lưới kết nối rộng lớn**

Zigbee sở hữu lên đến 65.000 điểm nút giúp nó trở thành một mạng lưới kết nối lớn hoạt động một cách hiệu quả và ổn định. Trong đó, mỗi điểm nút đều có vai trò truyền tải thông tin độc lập, chúng tự kết nối và truyền tải dữ liệu với nhau mà không cần kết nối một cách liên tục với bộ điều khiển trung tâm. Nếu có một nút bị lỗi thì nút điều khiển sẽ tự động kết nối để đảm bảo mạng lưới có thể hoạt động một cách hiệu quả.

**Sử dụng năng lượng hiệu quả**

Khi có nhiều thiết bị kết nối trong ngôi nhà thông minh, việc tiêu tốn điện năng khiến nhiều người lo ngại. Zigbee sẽ giúp người dùng sử dụng nguồn điện một cách hiệu quả, không những thế nó còn giúp người dùng kiểm soát tốt mọi hoạt động của các thiết bị nhờ khả năng hẹn lịch cho các thiết bị hoạt động. Do đó, điện năng tiêu tốn cho các thiết bị chuẩn Zigbee thường rất tiết kiệm.

**Phạm vi tín hiệu**

Zigbee có thể hoạt động hiệu quả trong nhà thông minh với phạm vi kết nối trong khoảng 10-20 mét bởi nó sử dụng ít năng lượng hơn.

**Khả năng tương thích cao**

Hiện nay, mạng Zigbee được hỗ trợ trên rất nhiều thiết bị, nó có khả năng tương thích với đa số các thiết bị thông minh được sử dụng cho ngôi nhà.

**Độ tin cậy**

Zigbee là công nghệ không dây hoạt động tốt trong ngôi nhà thông minh, cũng nhờ đó, độ tin cậy của mạng Zigbee khá cao được nhiều người tin dùng .

1. **Ưu nhược điểm khi dùng thiết bị thông minh Zigbee**

Có thể thấy Zigbee là một công nghệ không dây cung cấp giải pháp cho việc điều khiển các thiết bị trong cuộc sống hiện đại. Nhìn chung, Zigbee có một số ưu nhược điểm như sau:

**Ưu điểm của mạng Zigbee**

Hiện nay, mạng Zigbee được dùng rộng rãi trong nhà thông minh bởi nó có những ưu điểm vượt trội như:

* Lắp đặt đơn giản: Việc lắp đặt các thiết bị có sử dụng mạng Zigbee rất đơn giản và dễ dàng.
* Kết nối internet: Thông qua kết nối internet người dùng dễ dàng điều khiển các thiết bị. Nhờ đó việc điều khiển trong ngôi nhà thông minh của người dùng cũng dễ dàng hơn.
* Khả năng mở rộng cao: Mạng Zigbee không cần nhiều cục phát trung tâm bởi vì, các thiết bị sử dụng mạng này vừa có thể nhận vừa có thể truyền dữ liệu cho các thiết bị khác. Nhờ vậy mà nó sẽ trở thành một mạng lưới mở rộng cực lớn. Zigbee có thể mở rộng lên đến 65.000 thiết bị trong cùng hệ thống.
* Zigbee được sử dụng mã hóa AES-128, nhờ đó mà nó có độ bảo mật cao.
* Hoạt động ổn định: Đây là công nghệ mạng hoạt động ổn định trong phạm vi có mật độ tín hiệu dày đặc.

**Nhược điểm của ZigBee**

Ngoài những ưu điểm trên thì Zigbee có một số nhược điểm như:

* Trong một phạm vi có diện tích quá rộng thì mạng Zigbee không thể phủ sóng hết toàn bộ. Lúc này sẽ cần một thiết giúp mở rộng độ phủ sóng là ZigBee Repeater.
* Mạng Zigbee sẽ bị yếu đi khi gặp vật cản lớn hoặc nhà có nhiều phòng. Đặc biệt nó không thể xuyên tường được.
* Độ ổn định của sóng Zigbee không ổn định bằng thiết bị đi dây.

1. **Ứng dụng của Zigbee đối với các thiết bị IoT**

### 6.1 Nhà thông minh (Smart Home)

Zigbee được sử dụng để kết nối các thiết bị thông minh trong nhà, bao gồm:

* Công tắc, ổ cắm thông minh: Điều khiển từ xa qua điện thoại.
* Hệ thống chiếu sáng thông minh: Bóng đèn có thể điều chỉnh độ sáng, đổi màu, hoặc bật/tắt tự động.
* Cảm biến nhiệt độ, độ ẩm, chuyển động: Tích hợp vào hệ thống điều hòa hoặc cảnh báo xâm nhập.
* Khóa cửa thông minh: Mở cửa bằng vân tay, mã PIN, hoặc điều khiển từ xa.



**Hình 4. Nhà thông minh sử dụng giao thức Zigbee với các thiết bị Iot**

### 6.2 Thành phố thông minh (Smart City)

Zigbee giúp kết nối và quản lý:

* Hệ thống đèn đường thông minh: Tự động điều chỉnh độ sáng theo môi trường, tiết kiệm năng lượng.
* Bãi đỗ xe thông minh: Cảm biến phát hiện chỗ trống và hướng dẫn tài xế.
* Giám sát môi trường: Cảm biến đo mức độ ô nhiễm không khí, tiếng ồn, độ ẩm.

c. Công nghiệp (Industrial IoT - IIoT)

Zigbee hỗ trợ tự động hóa trong sản xuất và giám sát môi trường nhà máy:

* Giám sát dây chuyền sản xuất: Thu thập dữ liệu từ cảm biến để tối ưu quy trình.
* Hệ thống cảnh báo rò rỉ khí, nhiệt độ cao: Đảm bảo an toàn lao động.
* Kiểm soát năng lượng: Quản lý mức tiêu thụ điện trong nhà máy.

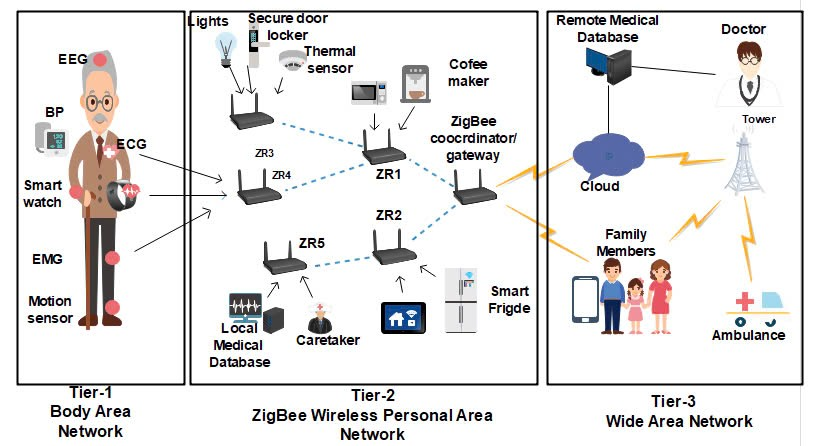


**Hình 5. Thành phố thông minh sử dụng giao thức Zigbee**

### 6.3 Y tế thông minh (Smart Healthcare)

Zigbee giúp kết nối thiết bị y tế để theo dõi sức khỏe từ xa:

* Máy đo huyết áp, nhịp tim, nhiệt độ: Truyền dữ liệu đến bác sĩ hoặc trung tâm y tế.
* Cảm biến giám sát bệnh nhân: Theo dõi vị trí và tình trạng bệnh nhân trong bệnh viện.

****

**Hình 6. Y tế thông minh**

1. **SO SÁNH Zigbee với các công nghệ mạng không dây khác**

Bảng 1. So sánh các công nghệ mạng không dây

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **Zigbee (Nhà thông minh, IoT)** | **Wi-Fi (Mạng không dây, Internet)** | **Bluetooth (Kết nối thiết bị cá nhân)** |
| **Tần số hoạt động** | 2.4 GHz, 915 MHz, 868 MHz | 2.4 GHz, 5 GHz, 6 GHz | 2.4 GHz |
| **Tốc độ truyền dữ liệu** | 20 - 250 Kbps (tốc độ thấp, đủ cho cảm biến và điều khiển thiết bị) | 11 Mbps - vài Gbps (tốc độ cao, phù hợp truyền video, dữ liệu lớn) | 1 Mbps - 3 Mbps (tốc độ thấp, đủ để truyền âm thanh và dữ liệu nhỏ) |
| **Phạm vi hoạt động** | 10 - 100 mét (phụ thuộc vào môi trường và công suất phát) | 30 - 100 mét (có thể xa hơn nếu dùng bộ mở rộng sóng) | 1 - 10 mét (Bluetooth Classic), có thể lên đến 100 mét với Bluetooth Low Energy (BLE) |
| **Tiêu thụ năng lượng** | Rất thấp (thiết bị có thể chạy bằng pin trong nhiều năm) | Cao (cần nguồn điện liên tục, tiêu hao năng lượng lớn) | Thấp đến trung bình (BLE tiết kiệm pin hơn, phù hợp cho thiết bị đeo) |
| **Mô hình kết nối** | Hỗ trợ mạng lưới (Mesh) và kết nối điểm-điểm (Point-to-Point) | Mô hình hình sao (Star), tất cả thiết bị kết nối về một router trung tâm | Hỗ trợ kết nối điểm-điểm (Point-to-Point) và mạng lưới (Mesh với BLE) |
| **Độ trễ (Latency)** | Cao (vài trăm mili-giây, phù hợp cho điều khiển không thời gian thực) | Thấp (dưới 10 mili-giây, phù hợp truyền trực tiếp và chơi game) | Trung bình (BLE có độ trễ thấp hơn, thích hợp cho kết nối nhanh) |
| **Khả năng mở rộng** | Rất cao (hỗ trợ hàng nghìn thiết bị trong một mạng lưới) | Trung bình (bị giới hạn bởi số lượng thiết bị có thể kết nối vào router) | Thấp (kết nối tối đa chỉ vài thiết bị cùng lúc) |
| **Ứng dụng chính** | Nhà thông minh, cảm biến, tự động hóa, IoT, điều khiển thiết bị từ xa | Truy cập Internet, truyền video, họp trực tuyến, chơi game trực tuyến | Kết nối tai nghe, bàn phím, chuột, đồng hồ thông minh, truyền dữ liệu giữa điện thoại và máy tính |
| **Bảo mật** | Sử dụng mã hóa AES-128 để bảo vệ dữ liệu truyền đi | Hỗ trợ các chuẩn bảo mật WPA2, WPA3 để mã hóa dữ liệu | Thấp đến trung bình (thiết bị có sẵn trên điện thoại, laptop, chi phí thấp) |
| **Chi phí triển khai** | Thấp (thiết bị Zigbee có giá rẻ hơn so với Wi-Fi) | Cao (yêu cầu router, bộ phát Wi-Fi, và đường truyền Internet) | Thấp đến trung bình (thiết bị có sẵn trên điện thoại, laptop, chi phí thấp) |

1. **Công cụ KillerBee**

KillerBee là một bộ công cụ chuyên dụng để tấn công và kiểm tra bảo mật mạng Zigbee. Nó được thiết kế để hỗ trợ các chuyên gia bảo mật kiểm tra tính an toàn của các thiết bị và giao thức Zigbee trong hệ thống IoT, nhà thông minh và các ứng dụng công nghiệp.

🔹 Tính năng chính của KillerBee:

* Thu thập gói tin (Packet Sniffing): Nghe lén và phân tích dữ liệu Zigbee trong không gian mạng.
* Tấn công phát lại (Replay Attack): Thu và phát lại gói tin để kiểm tra tính bảo mật.
* Tấn công từ chối dịch vụ (DoS): Làm gián đoạn truyền thông trong mạng Zigbee.
* Giải mã dữ liệu: Phân tích và bẻ khóa dữ liệu Zigbee nếu có khóa mã hóa.
* Kết hợp với phần cứng: Hỗ trợ nhiều thiết bị như HackRF, Atmel RZ Raven, GoodFET, và Ubertooth.

🔹 Ứng dụng của KillerBee:

* Kiểm tra bảo mật các thiết bị Zigbee trong nhà thông minh, hệ thống cảm biến, IoT.
* Đánh giá lỗ hổng của mạng không dây dựa trên Zigbee.
* Hỗ trợ nghiên cứu bảo mật giao thức IEEE 802.15.4.

1. **Công cụ Zigator**

Zigator là một công cụ tập trung vào phân tích và khai thác lỗ hổng Zigbee, giúp kiểm tra an ninh của các thiết bị sử dụng giao thức này.

🔹 Tính năng chính của Zigator:

* Tấn công MITM (Man-in-the-Middle): Chặn và thay đổi dữ liệu truyền giữa các thiết bị Zigbee.
* Tấn công khóa mạng (Network Key Extraction): Trích xuất khóa mã hóa để giải mã gói tin.
* Giả mạo thiết bị (Device Impersonation): Giả lập một thiết bị hợp lệ để tấn công mạng Zigbee.
* Phân tích gói tin: Giúp nghiên cứu giao thức và tìm lỗ hổng trong việc triển khai Zigbee.

🔹 Ứng dụng của Zigator:

* Kiểm tra độ an toàn của các thiết bị Zigbee trong môi trường thực tế.
* Hỗ trợ nghiên cứu các phương pháp tấn công và phòng thủ đối với mạng Zigbee.
* Giúp các chuyên gia bảo mật kiểm tra khả năng chống lại tấn công của hệ thống Zigbee.

## 10. Công cụ Scapy

Scapy là một công cụ mạnh mẽ để tạo, phân tích và thao tác gói tin trên nhiều giao thức mạng, bao gồm Zigbee (IEEE 802.15.4). Công cụ này thường được sử dụng trong nghiên cứu bảo mật, kiểm thử xâm nhập và phân tích mạng.

🔹 **Tính năng chính của Scapy:**

* **Tạo và gửi gói tin tùy chỉnh:** Hỗ trợ nhiều giao thức, bao gồm Zigbee, TCP/IP, ICMP, ARP.
* **Sniffing gói tin:** Bắt và phân tích gói tin trực tiếp từ mạng.
* **Tấn công replay:** Thu và phát lại gói tin để kiểm tra tính bảo mật.
* **Tấn công Man-in-the-Middle (MITM):** Giả mạo và sửa đổi dữ liệu truyền giữa các thiết bị.
* **Khai thác lỗ hổng:** Kiểm tra tính an toàn của các thiết bị và giao thức.
* **Tích hợp với các công cụ khác:** Có thể sử dụng kết hợp với KillerBee, Zigator và Wireshark.

🔹 **Ứng dụng của Scapy:**

* **Phân tích mạng:** Xây dựng và kiểm thử các giao thức tùy chỉnh.
* **Kiểm thử bảo mật:** Phát hiện lỗ hổng trong các hệ thống mạng.
* **Giả lập tấn công:** Kiểm tra khả năng phòng thủ của hệ thống.
* **Nghiên cứu bảo mật Zigbee:** Hỗ trợ thao tác gói tin Zigbee để phân tích và khai thác lỗ hổng.

**11. So sánh các công cụ KillerBee, Zigator và Scapy**

Bảng 2. So sánh các công cụ KillerBee, Zigator và Scapy

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **KillerBee** | **Zigator** | **Scapy** |
| **Mục đích chính** | Kiểm thử bảo mật, phân tích giao thức Zigbee | Tấn công nâng cao, giả mạo thiết bị Zigbee | Phân tích, tạo, sửa đổi gói tin trên nhiều giao thức |
| **Hỗ trợ giao thức** | Zigbee | Zigbee | Zigbee, TCP/IP, ICMP, ARP, IPv6, v.v. |
| **Hỗ trợ phần cứng** | RZUSBstick, Atmel RZ Raven, TelosB | Các thiết bị Zigbee mở rộng | Không yêu cầu phần cứng chuyên dụng |
| **Sniffing gói tin** | Có | Có | Có |
| **Tấn công replay (Replay Attack)** | Có | Có | Có |
| **Tấn công Man-in-the-Middle (MITM)** | Không | Có | Có |
| **Tấn công bẻ khóa AES-128** | Hạn chế | Mạnh | Hỗ trợ nhưng không chuyên sâu |
| **Tấn công từ chối dịch vụ (DoS)** | Có | Có | Có |
| **Giả mạo thiết bị Zigbee** | Không | Có | Có thể giả lập nhưng không chuyên biệt |
| **Tấn công giao thức mạng** | Zigbee | Zigbee | Zigbee, TCP/IP, ICMP, v.v. |
| **Tấn công dựa trên Injection** | Có, nhưng hạn chế | Có, mạnh hơn | Có, tùy chỉnh mạnh mẽ |
| **Hỗ trợ phân tích mạng** | Có, nhưng cơ bản | Có, chuyên sâu hơn | Có, mạnh mẽ |
| **Hỗ trợ tấn công brute-force key** | Có, nhưng giới hạn | Có, mạnh hơn và hiệu quả hơn | Không chuyên biệt cho brute-force key |
| **Giao diện người dùng** | Command-line | Command-line | Command-line (Python API) |
| **Tích hợp công cụ khác** | Có thể kết hợp với Wireshark, Scapy | Hỗ trợ tích hợp với các công cụ nâng cao | Wireshark, KillerBee, Zigator |

# CHƯƠNG 3 : PHƯƠNG PHÁP THỰC HIỆN

**1. Tổng quan hệ thống**

Triển khai Scapy trong môi trường của python trên Ubuntu sử dụng thiết bị ảo giúp nghiên cứu và thử nghiệm bảo mật ZigBee mà không cần phần cứng vật lý. Cách tiếp cận này phù hợp cho môi trường học tập, huấn luyện và phát triển các kỹ thuật phân tích mạng ZigBee.

**Ưu điểm của việc sử dụng thiết bị ảo:**

* Không cần đầu tư vào phần cứng ZigBee chuyên dụng
* Dễ dàng thiết lập môi trường thử nghiệm đồng nhất
* Có thể mô phỏng nhiều topo mạng và kịch bản khác nhau
* Thuận lợi cho mục đích giáo dục và nghiên cứu

**2. Yêu cầu hệ thống**

**a. Cấu hình máy chủ Ubuntu**

* Ubuntu 20.04 LTS hoặc mới hơn
* RAM: tối thiểu 8GB (khuyến nghị 16GB)
* CPU: 4 cores trở lên
* Ổ cứng: 50GB trống
* Quyền sudo/root

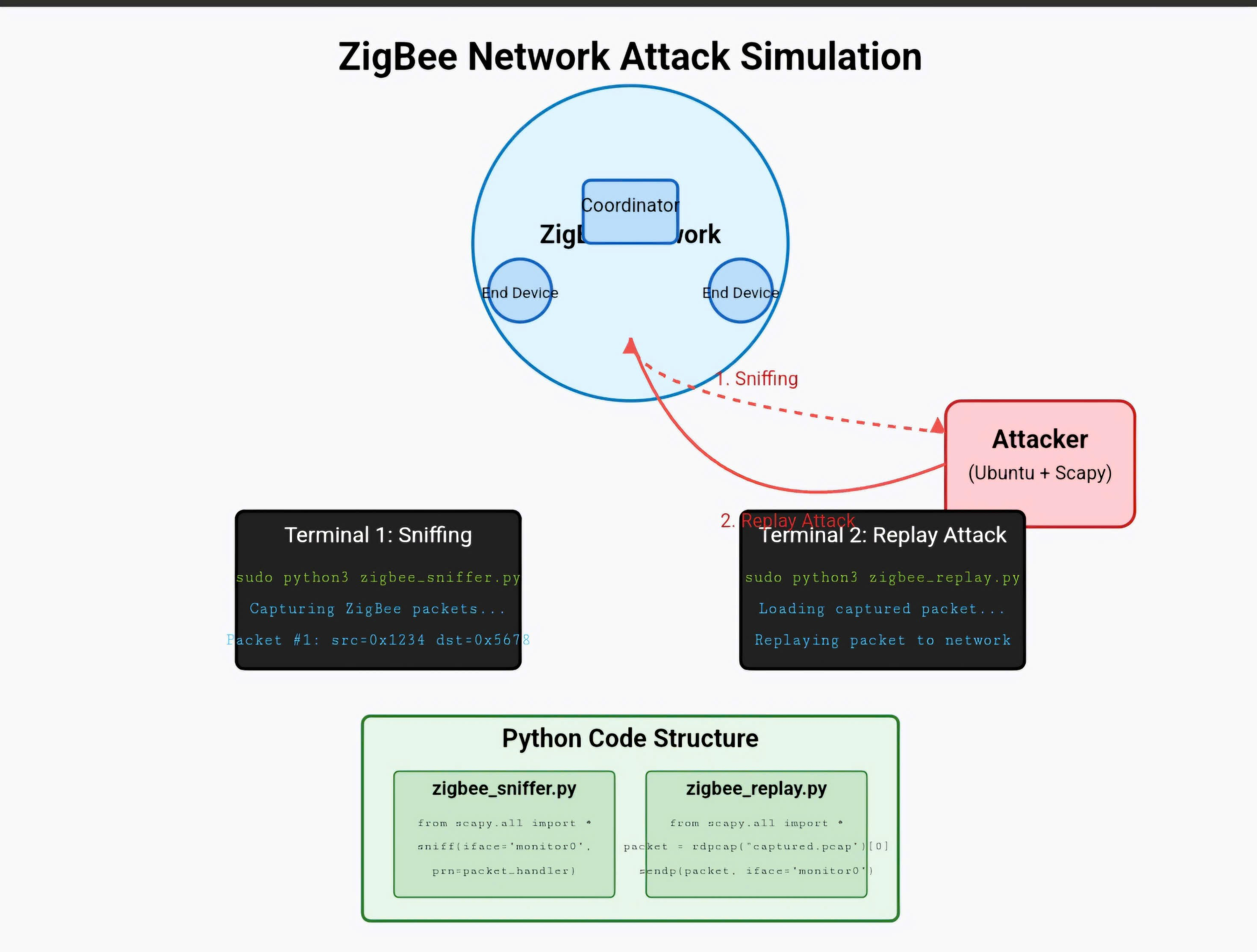
**b. Phần mềm cần thiết**

* QEMU/KVM hoặc VirtualBox
* GNS3 (để mô phỏng mạng)
* Python 3.8+ và pip
* Git
* Wireshark
* Phần mềm VMWare( dùng để chạy hệ điều hành ảo)

**3. Kiến trúc hệ thống**

### a) Hệ thống bao gồm các thành phần chính:

1. **Máy tính vật lý:** dùng để mở các máy ảo để thực hiện cuộc tấn công
2. **VMWare Workstation**: phần mềm dùng để chạy các hệ điều hành ảo mà không tốn nhiều dung lượng trên máy vật lý
3. **Ubuntu VM**: Nền tảng cơ sở chạy tất cả máy ảo và dịch vụ
4. **Python Container**: Chứa bộ công cụ Scapy trong môi trường container
5. **Zigbee emulation**:phần mềm giả lập(mô phỏng) mạng zigbee và các thiết bị ziggbee trong môi trường ảo



**Hình 7. Sơ đồ mạng**

**Cứ dựa theo sơ đồ thì quy trình hoạt động của nó :**

* **Máy tính vật lý (Host PC)** là nền tảng cơ bản với phần cứng thực tế (CPU Intel Core i7, RAM 16GB DDR4, ổ cứng 1TB SSD) và chạy Windows 10/11.
* **Trên máy tính vật lý**, người dùng tạo và chạy một **máy ảo Ubuntu** thông qua phần mềm ảo hóa (như VirtualBox hoặc VMware - không được chỉ rõ trong hình). Máy ảo này được cấp phát tài nguyên từ máy vật lý: 4 cores CPU, 8GB RAM, 40GB bộ nhớ.
* **Trong máy ảo Ubuntu**, có hai thành phần chính hoạt động song song: a) **Python Container chứa Scapy Tools**:
* Docker tạo môi trường cô lập để chạy các công cụ scapy
* Các công cụ này bao gồm:
  + zbstumbler: Dùng để tìm kiếm và phát hiện các mạng ZigBee
  + zbdump/zbconvert: Thu thập và chuyển đổi dữ liệu gói tin ZigBee
  + zbgoodfind/zbassocflood: Phân tích và thử nghiệm tấn công vào mạng ZigBee
  + rfmon + scapy: Thiết lập chế độ giám sát và phân tích gói tin

### b) ZigBee Emulator:

* Tạo một môi trường giả lập hoàn chỉnh cho mạng ZigBee
* Gồm các thành phần ảo hóa:
  + Virtual Coordinator: Điều phối viên ảo - thiết bị trung tâm điều khiển mạng ZigBee
  + Virtual Router: Bộ định tuyến ảo - chuyển tiếp dữ liệu trong mạng ZigBee
  + Virtual End Devices: Các thiết bị đầu cuối ảo - cảm biến, công tắc, v.v.
  + Cấu hình trên kênh 15 với PAN ID: 0x1234

### c) Quy trình hoạt động:

* Các công cụ scapy trong container Python tương tác với môi trường ZigBee Emulator
* Dữ liệu được trao đổi giữa hai thành phần thông qua mạng ảo
* Docker Container có thể gửi các lệnh, thu thập và phân tích gói tin từ môi trường ZigBee giả lập
* Các thiết bị ZigBee ảo trong Emulator tạo thành một mạng hoàn chỉnh với các chức năng tương tự như một mạng ZigBee thật

### d) Kết nối với máy chủ:

* Máy ảo Ubuntu được cấu hình với mạng NAT & Host-only:
  + NAT cho phép máy ảo kết nối internet thông qua máy vật lý
  + Host-only tạo mạng cô lập giữa máy ảo và máy vật lý để truyền dữ liệu an toàn

### e) Hiệu năng đồ họa:

* Máy ảo Ubuntu có 3D acceleration được bật để đảm bảo hiệu năng đồ họa tốt
* USB passthrough cho phép các thiết bị USB thực (như dongles ZigBee) được kết nối trực tiếp với máy ảo

# CHƯƠNG 4 : TRIỂN KHAI

* 1. **KillerBee**

### Bước 1: Cài đặt các gói cần thiết

Cập nhật hệ thống và cài các gói cần thiết trước khi cài KillerBee:

**A screenshot of a computer

AI-generated content may be incorrect.**

**Hình 8. Cập nhật hệ thống**

*sudo apt install python3-pip python3-setuptools python3-dev libgcrypt20-dev libusb-1.0-0-dev -y*

* **python3-pip**: Công cụ quản lý thư viện Python.
* **python3-setuptools**: Hỗ trợ cài đặt các gói Python.
* **python3-dev**: Bao gồm các tập tin cần thiết để biên dịch phần mở rộng Python.
* **libgcrypt20-dev**: Thư viện mã hóa cần thiết cho KillerBee.
* **libusb-1.0-0-dev**: Hỗ trợ giao tiếp với thiết bị USB (quan trọng cho việc sử dụng adapter Zigbee).

**A computer screen shot of a program

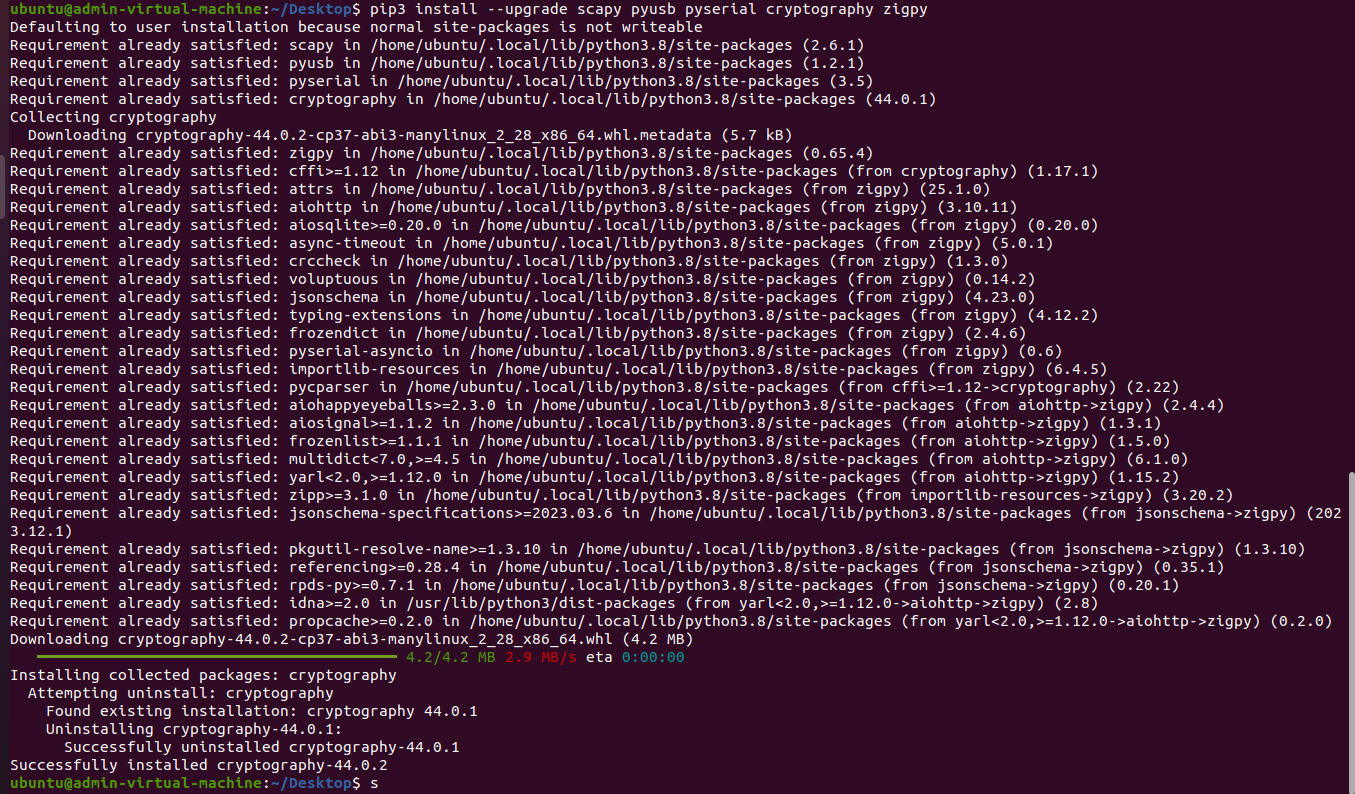
AI-generated content may be incorrect.**

**Hình 9. Công cụ quản lí thư viện python3**

### Bước 2: Cài đặt thư viện phụ thuộc

*pip3 install --upgrade scapy pyusb pyserial cryptography zigpy*

* **scapy**: Công cụ thao tác gói tin, cần thiết cho phân tích Zigbee.
* **pyusb**: Cho phép Python giao tiếp với thiết bị USB.
* **pyserial**: Hỗ trợ giao tiếp với thiết bị qua cổng Serial.
* **cryptography**: Cung cấp các thuật toán mã hóa, quan trọng cho việc phân tích Zigbee.
* **zigpy**: Một thư viện Python để làm việc với Zigbee.
* **--upgrade**: Đảm bảo thư viện được cập nhật lên phiên bản mới nhất.

****

**Hình 10. Cài đặt các thư viện phụ thuộc**

### Bước 3: Tải KillerBee từ GitHub

**Clone KillerBee từ GitHub:**

*git clone* [*https://github.com/riverloopsec/killerbee.git*](https://github.com/riverloopsec/killerbee.git)

* **git clone**: Sao chép (clone) mã nguồn của KillerBee từ GitHub về máy.
* **https://github.com/riverloopsec/killerbee.git**: URL kho chứa mã nguồn của KillerBee.



**Hình 11. Clone KillerBee**

### Bước 4: Cài đặt KillerBee

*sudo python3 setup.py install*

* **python3 setup.py install**: Cài đặt KillerBee vào hệ thống.
* **sudo**: Đảm bảo có quyền cài đặt.

**A screen shot of a computer screen

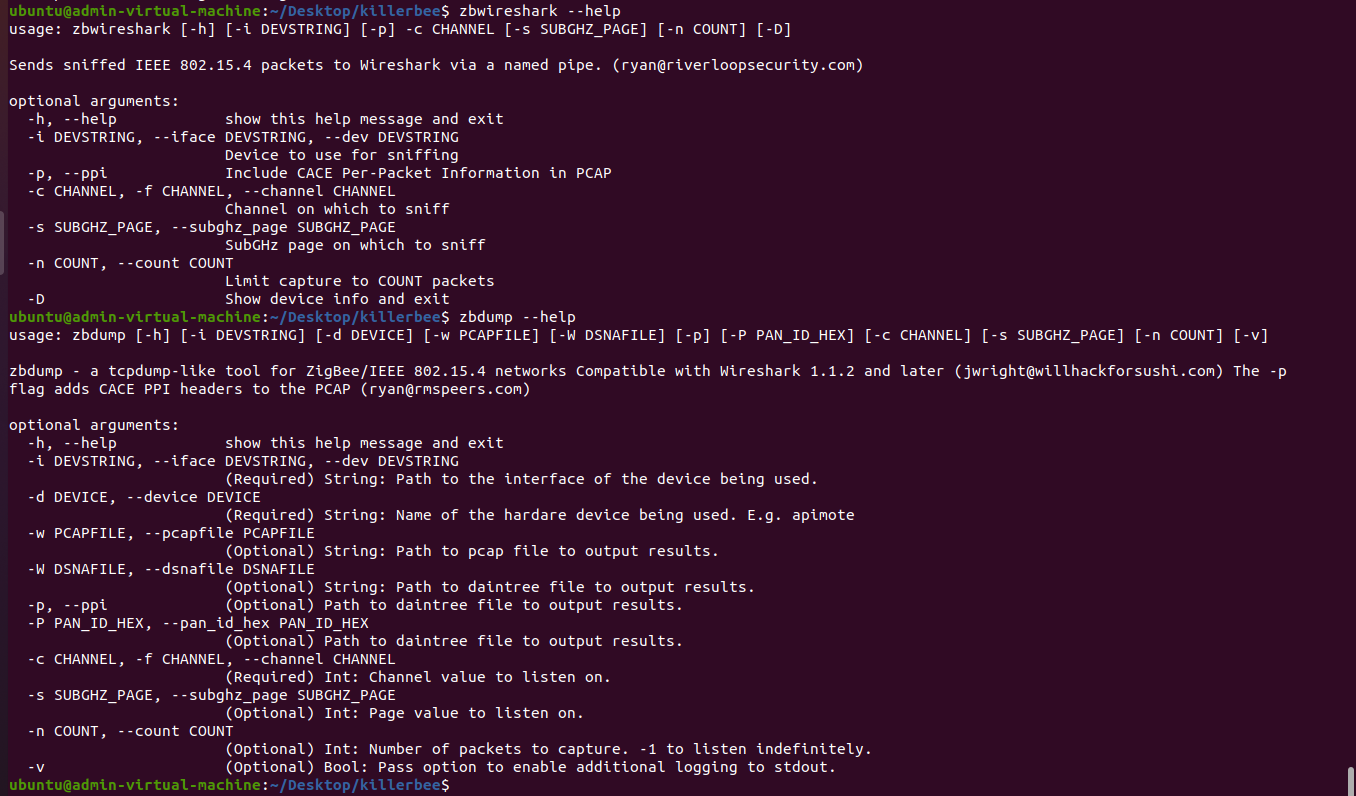
AI-generated content may be incorrect.**

**Hình 12. Cài đặt KillerBee**

### Bước 5: Kiểm tra KillerBee đã cài đặt thành công

zbwireshark --help

zbdump --help

****

**Hình 13. Kiểm tra KillerBee cài đặt thành công**

* 1. **Zigator**

*deactivate # Thoát môi trường ảo hiện tại*

*python3 -m venv zigator\_env\_new # Tạo môi trường ảo mới*

*source zigator\_env\_new/bin/activate # Kích hoạt môi trường mới*

*pip install --upgrade pip setuptools wheel*

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Hình 14. Tạo môi trường ảo hóa mới**

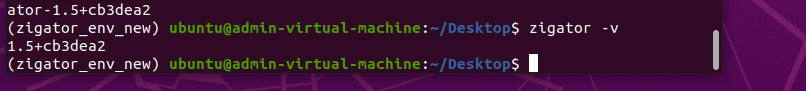
*pip install git+https://github.com/akestoridis/zigator.git*

**A computer screen shot of a program

AI-generated content may be incorrect.**

**Hình 15. Cài đặt Zigator**

*zigator -v*

****

**Hình 16. Kiểm tra cài đặt Zigator**

* 1. **Scapy**

### ****Bước 1: Clone repository Scapy từ GitHub****

*git clone* [*https://github.com/secdev/scapy.git*](https://github.com/secdev/scapy.git)

### Bước 2: Di chuyển vào thư mục Scapy

*cd scapy*

### Bước 3: Cài đặt Scapy bằng setup.py

*sudo python3 setup.py install*

*A screenshot of a computer screen

AI-generated content may be incorrect.*

**Hình 17. Cài đặt Scapy**

### Bước 4: Kiểm tra cài đặt Scapy

*python3 -c "from scapy.all import \*; print('Scapy đã cài đặt thành công\!')"*

*pip3 list | grep scapy*

*which python3*

*which pip3*

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Hình 18. Kiểm tra cài đặt Scapy**

## 4.Demo bằng công cụ Scapy

**Sniffing Attack:** Bắt gói tin Zigbee trên giao diện lo.  
**Replay Attack:** Lưu và phát lại gói tin đã bắt được.  
**Không cần thiết bị Zigbee thật**, chỉ cần Scapy và giao diện lo.

### Bước 1: Kiểm tra giao diện mạng Loopback

*ip a | grep lo*

### Bước 2: Sniffing Attack - Bắt gói tin Zigbee

*from scapy.all import sniff*

*# Bắt 10 gói tin trên giao diện loopback*

*pkts = sniff(iface="lo", count=10)*

*# Hiển thị các gói tin bắt được*

*pkts.show()*

### Bước 3: Gửi gói tin Zigbee giả lập

*from scapy.all import sendp*

*from scapy.layers.dot15d4 import Dot15d4*

*from scapy.layers.zigbee import ZigbeeNWK, ZigbeeAppDataPayload*

*pkt = Dot15d4() / ZigbeeNWK() / ZigbeeAppDataPayload()*

*# Gửi 10 gói tin*

*for \_ in range(10):*

*sendp(pkt, iface="lo")*

### Bước 4: Lưu gói tin đã bắt được

*from scapy.all import wrpcap*

*# Lưu gói tin vào file PCAP để phân tích sau này*

*wrpcap("zigbee\_sniffed.pcap", pkts)*

*print("Đã lưu gói tin vào zigbee\_sniffed.pcap")*

### Bước 5: Phát lại gói tin - Replay Attack

*from scapy.all import rdpcap, sendp*

*# Đọc gói tin đã lưu*

*pkts = rdpcap("zigbee\_sniffed.pcap")*

*# Phát lại toàn bộ gói tin*

*sendp(pkts, iface="lo")*

*print("Đã phát lại gói tin Zigbee!")*

### Bước 6: Kiểm tra xem Replay Attack có thành công không

*from scapy.all import sniff*

*pkts = sniff(iface="lo", count=5)*

*pkts.show()*

## 5. Demo bảo mật hệ thống zigbee

### 5.1. Giới thiệu về Demo

Demo tương tác bảo mật ZigBee được xây dựng để mô phỏng và minh họa các nguyên tắc bảo mật cơ bản trong mạng ZigBee. Demo này giúp người dùng hiểu rõ hơn về cách thức hoạt động của bảo mật ZigBee, các lỗ hổng tiềm ẩn và các biện pháp bảo vệ thông qua các tình huống thực tế.

### 5.2. Cấu trúc và Thành phần của Demo

Demo được thiết kế với các thành phần chính sau:

#### **5.2.1 Lớp ZigBeePacket**

* **Chức năng**: Biểu diễn gói tin ZigBee trong mạng.
* **Thuộc tính chính**:
  + timestamp: Thời gian tạo gói tin
  + src\_addr: Địa chỉ nguồn (16-bit)
  + dst\_addr: Địa chỉ đích (16-bit)
  + pan\_id: ID của mạng ZigBee
  + payload: Dữ liệu người dùng
  + encrypted: Trạng thái mã hóa (True/False)
  + nonce: Giá trị dùng một lần cho mã hóa
  + encrypted\_data: Dữ liệu đã mã hóa

#### **5.2.2 Lớp ZigBeeNetwork**

* **Chức năng**: Mô phỏng mạng ZigBee với các thiết bị và truyền thông.
* **Phương thức chính**:
  + create\_packet(): Tạo gói tin mới (có hoặc không mã hóa)
  + set\_security(): Bật/tắt bảo mật và thiết lập khóa mạng

#### **5.2.3 Lớp ZigBeeAttacker**

* **Chức năng**: Mô phỏng kẻ tấn công có thể nghe lén và phân tích gói tin.
* **Phương thức chính**:
  + capture\_network\_traffic(): Bắt gói tin từ mạng
  + analyze\_packets(): Phân tích các gói tin đã bắt được
  + try\_decrypt\_packet(): Thử giải mã gói tin với các khóa đã biết
  + recommend\_actions(): Đưa ra khuyến nghị bảo mật dựa trên phân tích

#### **5.2.4. Lớp ZigBeeSecurityAdvisor**

* **Chức năng**: Cung cấp tư vấn và triển khai bảo mật cho mạng.
* **Phương thức chính**:
  + generate\_strong\_key(): Tạo khóa mạng mạnh
  + implement\_security(): Triển khai các biện pháp bảo mật
  + security\_recommendations(): Đưa ra các khuyến nghị bảo mật nâng cao

### 5.3 Kỹ Thuật Mã Hóa Trong Demo

#### **5.3.1 Mã hóa AES-128 (CTR Mode)**

* **Thuật toán**: Advanced Encryption Standard với khóa 128-bit
* **Chế độ hoạt động**: Counter (CTR) mode
* **Quá trình mã hóa**:
  1. Tạo nonce ngẫu nhiên 12 byte
  2. Khởi tạo counter từ nonce
  3. Tạo cipher với khóa mạng
  4. Mã hóa dữ liệu
  5. Lưu nonce và dữ liệu đã mã hóa vào gói tin

#### **5.3.2 Quá trình giải mã**

* **Quá trình**:
  1. Trích xuất nonce từ gói tin
  2. Khởi tạo lại counter từ nonce
  3. Tạo cipher với khóa mạng
  4. Giải mã dữ liệu
  5. Kiểm tra tính hợp lệ của dữ liệu sau giải mã

### 5.4 Các Biện Pháp Bảo Mật Được Triển Khai

#### **5.4.1 Mã hóa dữ liệu**

* **Triển khai**: Sử dụng mã hóa AES-128 cho tất cả gói tin
* **Hiệu quả**: Ngăn chặn việc truy cập trực tiếp vào nội dung dữ liệu

#### **5.4.2 Quản lý khóa**

* **Triển khai**: Tạo và sử dụng khóa mạng mạnh (ngẫu nhiên 128-bit)
* **Hiệu quả**: Tránh sử dụng khóa mặc định dễ bị đoán

#### **5.4.3 Các khuyến nghị bảo mật nâng cao**

* **Triển khai**: Hiển thị và giải thích các biện pháp bảo mật toàn diện
* **Hiệu quả**: Nâng cao nhận thức và kiến thức về bảo mật ZigBee

**Link youtube**

**https://youtu.be/Umbco7NKYVU**

**https://youtu.be/LTc5TQt-68E**

**https://youtu.be/bVLzhsvqBy4**

# CHƯƠNG 5 : ĐÁNH GIÁ KẾT LUẬN

## 1.Đánh giá kết quả nghiên cứu

Sau khi thực hiện nghiên cứu và triển khai các cuộc tấn công sniffing và replay attack bằng Scapy trên Python 3 trong môi trường mô phỏng không sử dụng thiết bị thật, nhóm nghiên cứu đã đạt được những kết quả sau:

### 1.1Xác định lỗ hổng Zigbee:

* + Scapy đã chứng minh được khả năng sniffing các gói tin Zigbee trong môi trường mô phỏng.
  + Replay attack có thể thực hiện thành công, chứng minh tính dễ bị tấn công của hệ thống nếu không có biện pháp bảo vệ phù hợp.

### 1.2.Thực hiện các cuộc tấn công:

* + **Sniffing attack**: Sử dụng Scapy để bắt gói tin Zigbee trên giao diện loopback của hệ thống mô phỏng.

Screens screenshot of a computer

AI-generated content may be incorrect.

**Hình 19. Kết quả Sniffing**

* + **Replay attack**: Ghi lại và phát lại các gói tin đã bắt được để điều khiển thiết bị giả lập trong môi trường mô phỏng.

Screens screenshot of a computer

AI-generated content may be incorrect.

**Hình 20. kết quả Replay Attack**

### 1.3.Triển khai các biện pháp bảo vệ:

* + Cấu hình mã hóa AES-128 trên gói tin để tránh bị sniffing.
  + Áp dụng xác thực thiết bị để ngăn chặn replay attack.
  + Phát triển hệ thống phát hiện xâm nhập (IDS) để cảnh báo các hành vi bất thường.

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 21. kết quả bảo mật**

## 2. Hạn chế và hướng phát triển

Mặc dù đã đạt được những kết quả đáng kể, nghiên cứu vẫn tồn tại một số hạn chế:

* Chưa thử nghiệm trên thiết bị Zigbee thực tế, chỉ mới triển khai trên môi trường mô phỏng.
* Chưa đánh giá được mức độ ảnh hưởng của tấn công replay trong môi trường mạng Zigbee thực.

Hướng phát triển:

* Triển khai nghiên cứu trên thiết bị Zigbee thực tế để đánh giá chính xác mức độ ảnh hưởng.
* Kết hợp AI và Machine Learning để phát hiện và ngăn chặn tấn công.
* Phát triển giải pháp bảo mật dựa trên IDS để giám sát và phát hiện các cuộc tấn công tự động.

## 3. Kết luận

Qua nghiên cứu, nhóm đã chứng minh được rằng Zigbee có thể bị tấn công bởi sniffing và replay attack ngay cả trong môi trường mô phỏng. Các biện pháp bảo vệ như mã hóa dữ liệu, xác thực thiết bị và sử dụng hệ thống phát hiện xâm nhập có thể giúp giảm thiểu rủi ro.

Nghiên cứu này đóng góp vào việc nâng cao nhận thức về bảo mật Zigbee và cung cấp cơ sở cho các giải pháp bảo vệ hiệu quả hơn trong các hệ thống IoT.

* 1. **Tài liệu tham khảo**

. Chat GPT

. <https://hunonic.com/zigbe-la-gi/#Khac_biet_giua_Zigbee_va_Bluetooth_Mesh>

. <https://fptsmarthome.vn/kien-thuc/zigbee-la-gi?srsltid=AfmBOopdIwyx9TK-1qtETSCOwDv6yRwtgJAZxcC9m9bKIoHiqMs3e_xy>

. https://viettelidc.com.vn/tin-tuc/iot-la-gi-nhung-ung-dung-noi-bat-cua-iot