**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



A picture containing text, vector graphics, clipart

Description automatically generatedA picture containing graphics, clipart, logo, symbol

Description automatically generated

MÔN HỌC: HỆ THỐNG NHÚNG MẠNG KHÔNG DÂY

**BÁO CÁO ĐỒ ÁN**

**RaspAP Trên Raspberry pi 4**

**Debug quá trình kết nối mạng**

**Giảng viên hướng dẫn:**

**PGS.TS.LÊ TRUNG QUÂN**

**Sinh viên thực hiện:**

Nguyễn Đức Toàn – 22521490

Đặng Chí Thành – 22521344

Nguyễn Đặng Khánh Quốc - 22521212

**TP. HỒ CHÍ MINH, 2024**

**MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 2**](#_heading=h.30j0zll)

[1. Tổng quan về môn học 2](#_heading=h.1fob9te)

[2. Lý do chọn đề tài 2](#_heading=h.2et92p0)

[RaspAP là một công cụ mã nguồn mở phổ biến, được sử dụng để biến Raspberry Pi thành một Access Point (AP) không dây. Nghiên cứu RaspAP mang lại nhiều giá trị học thuật và thực tiễn như: 2](#_heading=h.tyjcwt)

[3. Ý nghĩa của đề tài 2](#_heading=h.3dy6vkm)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3**](#_heading=h.1t3h5sf)

[2.1 Tổng quan về hệ thống mạng không dây 3](#_heading=h.4d34og8)

[2.2 Khái niệm Access Point và ứng dụng trên Raspberry Pi 3](#_heading=h.2s8eyo1)

[2.3 Giới thiệu về RaspAP 3](#_heading=h.17dp8vu)

[2.4 Quá trình kết nối Wi-Fi giữa Client và Access Point 3](#_heading=h.3rdcrjn)

[2.4.1 Scanning (Quá trình quét và phát hiện AP) 4](#_heading=h.26in1rg)

[2.4.2 Authentication (Quá trình xác thực) 4](#_heading=h.lnxbz9)

[2.4.3 Association (Quá trình liên kết) 5](#_heading=h.35nkun2)

[**CHƯƠNG 3: QUY TRÌNH THỰC HIỆN VÀ CẤU HÌNH HỆ THỐNG 5**](#_heading=h.1ksv4uv)

[3.1 Mục tiêu của chương 6](#_heading=h.44sinio)

[3.2 Các bước cấu hình hệ thống 6](#_heading=h.2jxsxqh)

[**CHƯƠNG 4: DEBUG QUÁ TRÌNH HỆ THỐNG 13**](#_heading=h.1ci93xb)

[4.1 Mục tiêu của việc Debug 13](#_heading=h.3whwml4)

[4.2 Các công cụ và phương pháp Debug 13](#_heading=h.2bn6wsx)

[4.3 Quá trình phát beacon 13](#_heading=h.qsh70q)

[4.4 Quá trình nhận gói tin và xử lý sự kiện association 28](#_heading=h.3as4poj)

[4.5 Quá trình xác thực 34](#_heading=h.1pxezwc)

[**CHƯƠNG 5: KẾT LUẬN 53**](#_heading=h.49x2ik5)

[**TÀI LIỆU THAM KHẢO 53**](#_heading=h.147n2zr)

## CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

### 1. Tổng quan về môn học

Trong thời đại công nghệ không dây phát triển mạnh mẽ, hệ thống mạng không dây đã trở thành một phần thiết yếu trong cuộc sống hàng ngày và các ứng dụng công nghiệp. Đặc biệt, việc triển khai các hệ thống mạng không dây trên nền tảng thiết bị nhúng, như Raspberry Pi, đang mở ra nhiều hướng đi mới trong nghiên cứu và ứng dụng thực tế.

Môn học "Hệ thống nhúng Mạng không dây" cung cấp cho sinh viên nền tảng kiến thức và kỹ năng về thiết kế, cấu hình, và quản lý hệ thống mạng không dây. Đề tài "Phân tích quá trình kết nối không dây của RaspAP trên Raspberry Pi 4" được thực hiện nhằm cụ thể hóa và ứng dụng những kiến thức này vào thực tiễn.

### 2. Lý do chọn đề tài

### RaspAP là một công cụ mã nguồn mở phổ biến, được sử dụng để biến Raspberry Pi thành một Access Point (AP) không dây. Nghiên cứu RaspAP mang lại nhiều giá trị học thuật và thực tiễn như:

* Hiểu rõ cơ chế hoạt động của hệ thống: Nắm bắt cách các thành phần chính như hostapd, dnsmasq, và Lighttpd phối hợp để tạo nên một điểm truy cập không dây hoàn chỉnh.
* Phân tích quy trình kết nối: Làm rõ ba giai đoạn quan trọng trong kết nối Wi-Fi gồm Probing, Authentication, và Association, qua đó hiểu sâu hơn về vai trò của hostapd và cách các gói tin được trao đổi giữa client và AP.

### 3. Ý nghĩa của đề tài

**3.1. Giá trị thực tiễn**

Cung cấp kỹ năng triển khai và quản lý một hệ thống Access Point thực tế, có thể áp dụng vào các giải pháp mạng không dây cho gia đình, doanh nghiệp nhỏ, hoặc các môi trường nghiên cứu.

Góp phần hỗ trợ xây dựng các hệ thống mạng nhúng trong các ứng dụng như IoT, tự động hóa, hoặc mạng cảm biến không dây.

**3.2. Giá trị học thuật**

Làm rõ cách thức hoạt động của các công cụ mã nguồn mở trong mạng không dây.

Tạo nền tảng để nghiên cứu sâu hơn về tối ưu hóa hiệu năng mạng, bảo mật kết nối Wi-Fi, hoặc tích hợp các công nghệ hiện đại như Wi-Fi 6.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Tổng quan về hệ thống mạng không dây

Mạng không dây (Wireless Network) là loại mạng kết nối các thiết bị mà không cần đến dây cáp vật lý, sử dụng sóng vô tuyến để truyền dữ liệu. Hệ thống mạng không dây được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ sự tiện lợi và linh hoạt. Đặc biệt, các thiết bị nhúng nhỏ gọn như Raspberry Pi có thể trở thành các điểm truy cập mạng không dây (Access Point) để cung cấp kết nối cho các thiết bị khác.

### 2.2 Khái niệm Access Point và ứng dụng trên Raspberry Pi

Access Point (AP) là thiết bị cung cấp kết nối mạng không dây cho các thiết bị xung quanh, cho phép chúng truy cập vào mạng nội bộ hoặc internet. Khi sử dụng Raspberry Pi làm AccessPoint, thiết bị này có thể hoạt động như một router mini, cung cấp kết nối cho các thiết bị khác mà không cần thêm phần cứng ngoài. RaspAP là một giải pháp phổ biến giúp biến Raspberry Pi thành một AP dễ cấu hình, đi kèm giao diện web thân thiện và có nhiều tính năng cho người dùng.

### 2.3 Giới thiệu về RaspAP

RaspAP là một bộ công cụ mã nguồn mở, cho phép người dùng cấu hình Raspberry Pi làm Access Point một cách dễ dàng.

RaspAP hoạt động dựa trên ba thành phần chính:

* hostapd: Đảm nhiệm các chức năng liên quan đến bảo mật và chứng thực (authentication) giữa AP và client.
* dnsmasq: Cung cấp dịch vụ DHCP và DNS cho mạng nội bộ, giúp cấp phát địa chỉ IP động cho các thiết bị kết nối.
* Lighttpd: Web server để cung cấp giao diện quản lý cấu hình của RaspAP.

### 2.4 Quá trình kết nối Wi-Fi giữa Client và Access Point

Kết nối mạng không dây giữa client và Access Point (AP) trải qua ba giai đoạn chính, đó là Probing, Authentication, và Association. Mỗi giai đoạn đóng một vai trò quan trọng trong việc thiết lập một kết nối mạng an toàn và ổn định giữa hai thiết bị.

A diagram of a computer connection

Description automatically generated

### 2.4.1 Scanning (Quá trình quét và phát hiện AP)

**Giai đoạn Scanning** là bước đầu tiên trong quá trình kết nối không dây. Khi một client muốn kết nối với một mạng không dây, nó sẽ bắt đầu tìm kiếm các Access Point trong phạm vi. Quá trình này được thực hiện thông qua việc gửi các gói tin probe request hoặc nhận các gói beacon frame từ các AP xung quanh phát ra. Các gói tin không có thông tin cụ thể về mạng, mà chỉ là các thông tin cấu hình của AP.

**Probe Request:** Gói tin này bao gồm thông tin về SSID (tên mạng) mà client đang tìm kiếm, hoặc có thể được để trống nếu client muốn nhận thông tin từ tất cả các AP trong phạm vi. Khi AP nhận được gói tin probe request, nếu nó có thể hỗ trợ yêu cầu của client, nó sẽ trả lời với một gói tin probe response.

**Probe Response:** Gói tin này sẽ chứa các thông tin cơ bản về AP như SSID của mạng, các khả năng hỗ trợ (chẳng hạn như các kênh Wi-Fi, loại mã hóa, v.v.), và thông tin khác như chế độ bảo mật mà AP hỗ trợ (WPA2, WPA3, v.v.). Đây là thông tin mà client cần để quyết định có nên tiếp tục kết nối với AP hay không.

**Beacon Frame:** Là gói tin định kỳ do các AP phát ra, chứa thông tin về AP và mạng không dây. Thông tin trong beacon bao gồm: SSID, Channel, Tốc độ hỗ trợ, phương thức bảo mật, v.v.

Lúc này, client sẽ nhận được danh sách các AP có sẵn và có thể quyết định chọn AP phù hợp nhất để tiếp tục quá trình kết nối.

### 2.4.2 Authentication (Quá trình xác thực)

Sau khi client nhận được danh sách các AP có sẵn thông qua giai đoạn Probing, nếu nó chọn một AP để kết nối, bước tiếp theo là Authentication. Đây là quá trình mà client và AP xác nhận lẫn nhau để đảm bảo rằng client có quyền truy cập vào mạng. Các bước xác thực có thể khác nhau tùy thuộc vào phương thức bảo mật mà AP sử dụng.

**Authentication Request:** Khi client quyết định kết nối với một AP cụ thể, nó sẽ gửi một gói tin authentication request đến AP. Gói tin này thường chỉ đơn giản yêu cầu AP xác thực client mà không cần bất kỳ thông tin bảo mật nào.

**Authentication Response**: Nếu AP không yêu cầu bất kỳ bảo mật nào hoặc nếu quá trình xác thực được thực hiện thông qua các phương thức bảo mật như WPA2, thì client sẽ cần cung cấp một mật khẩu hoặc mã khóa bảo mật. AP sẽ xác nhận tính hợp lệ của thông tin bảo mật (chẳng hạn như mật khẩu đúng), và nếu thành công, quá trình xác thực sẽ được hoàn tất.

**Phương thức bảo mật:** Nếu AP sử dụng phương thức bảo mật như WPA2 hoặc WPA3, thì gói tin xác thực sẽ bao gồm một quá trình mã hóa khóa và việc trao đổi thông tin bảo mật giữa client và AP. Nếu thông tin bảo mật không hợp lệ (ví dụ, mật khẩu sai), quá trình kết nối sẽ bị gián đoạn và client sẽ không thể tiến hành kết nối.

Giai đoạn authentication giúp đảm bảo rằng chỉ các thiết bị có quyền truy cập hợp lệ mới có thể kết nối vào mạng, giúp bảo vệ mạng không dây khỏi sự truy cập trái phép.

### 2.4.3 Association (Quá trình liên kết)

Sau khi xác thực thành công, client sẽ gửi một gói tin association request đến AP để yêu cầu kết nối vào mạng. Đây là bước cuối cùng trong quá trình thiết lập kết nối Wi-Fi.

**Association Request:** Gói tin này chứa thông tin về client và khả năng của nó trong việc sử dụng các dịch vụ của AP. Nó có thể bao gồm thông tin về khả năng hỗ trợ kênh, loại bảo mật và các yêu cầu khác mà client có đối với AP. Gói tin association request giúp AP hiểu rõ về các yêu cầu của client và xác định xem liệu nó có thể cung cấp các dịch vụ cần thiết hay không.

**Association Response:** Nếu AP chấp nhận yêu cầu của client, nó sẽ gửi lại một gói tin association response. Gói tin này xác nhận rằng client đã được kết nối vào mạng và đã được cấp quyền truy cập vào các dịch vụ mạng. Sau khi nhận được gói tin association response, client sẽ có thể gửi và nhận dữ liệu từ AP như một phần của mạng không dây.

Trong giai đoạn Association, AP cấp phát cho client một địa chỉ IP (thông qua dịch vụ DHCP hoặc các phương thức khác), và client có thể bắt đầu giao tiếp với các thiết bị trong mạng nội bộ hoặc kết nối với internet, tùy thuộc vào cấu hình của AP.

## CHƯƠNG 3: QUY TRÌNH THỰC HIỆN VÀ CẤU HÌNH HỆ THỐNG

### 3.1 Mục tiêu của chương

Chương này tập trung vào quy trình cài đặt và cấu hình hệ thống Raspberry Pi với RaspAP để hoạt động như một Access Point, bao gồm các bước thiết lập cơ bản và kiểm tra chức năng. Các hướng dẫn chi tiết trong phần này sẽ giúp xây dựng một hệ thống mạng không dây ổn định cho phép client kết nối và truy cập vào mạng qua Raspberry Pi.

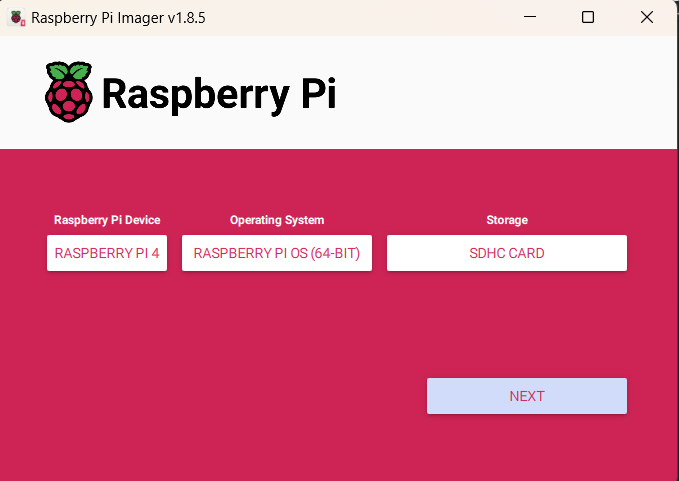
### Các bước cấu hình hệ thống

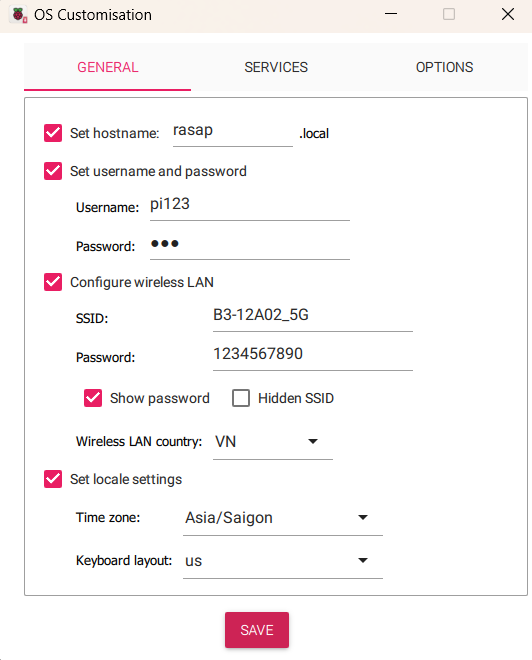
#### 3.2.1 Cài đặt hệ điều hành

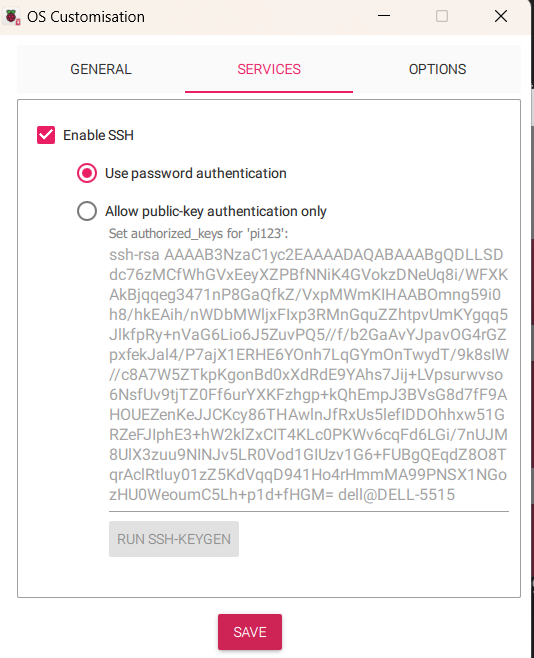
##### Bước 1: Chuẩn bị dụng cụ

* Raspberry Pi 4
* 1 micro usb và 1 đầu đọc thẻ
* 1 dây mạng LAN

##### Bước 2: Tải image vào thẻ micro usb sử dụng công cụ [Raspberry Pi Imager](https://www.raspberrypi.com/software/)







#### 3.2.2 Thiết lập kết nối tới raspberry để cấu hình

* Cách 1: Sử dụng putty để kết nối đến Raspberry Pi

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

* Cách 2: Sử dụng chính terminal để ssh đến Raspberry Pi

A screenshot of a computer

Description automatically generated

#### 3.2.3 Cài đặt RaspAP

Có hai cách cài RaspAP là cài đặt nhanh và cài đặt thủ công

* Cài đặt nhanh

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*Giao diện tải RaspAP*

* Cài đặt tay ([Manual installation](https://docs.raspap.com/manual/)):

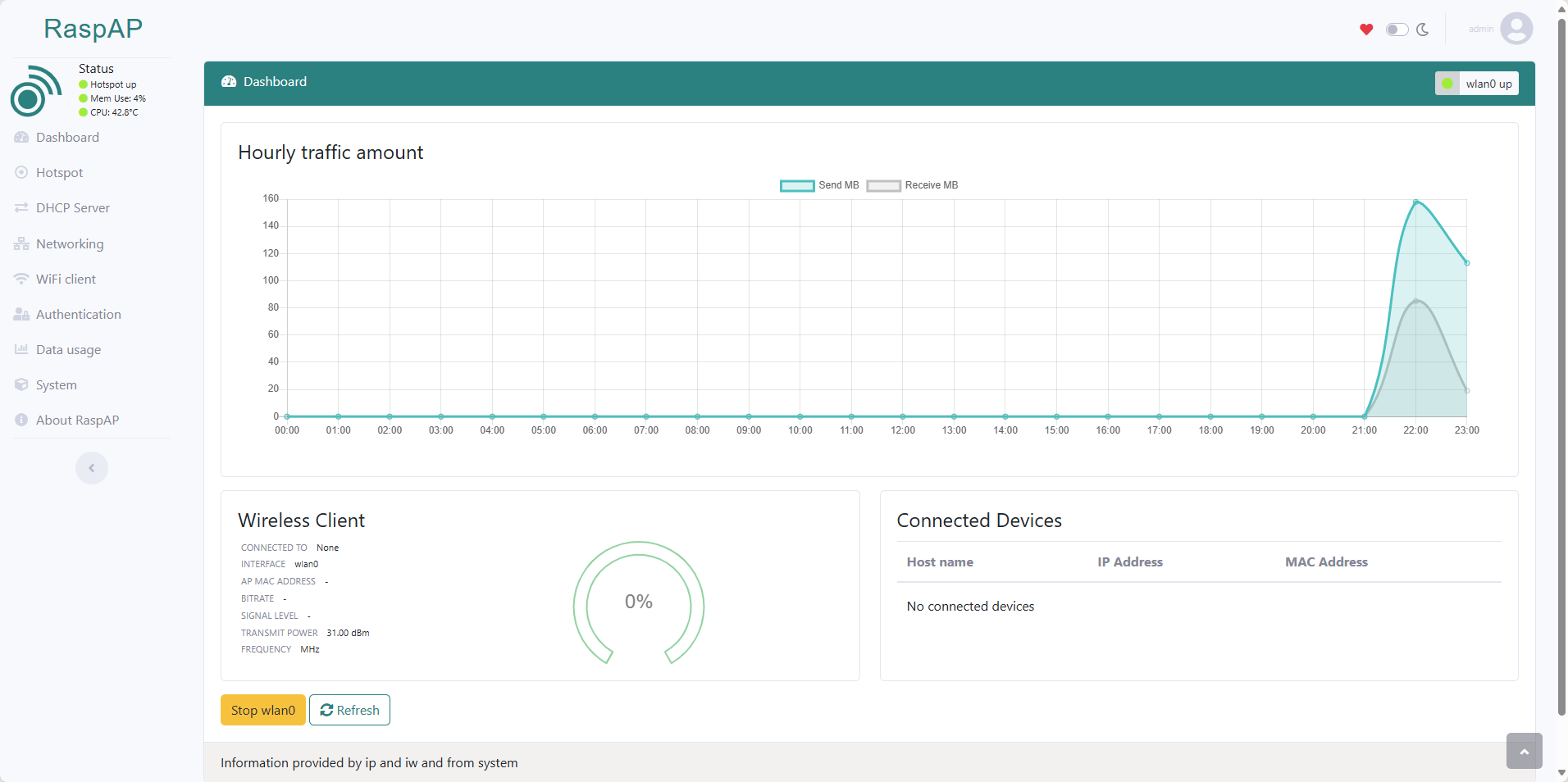
A screenshot of a computer

Description automatically generated

Sau khi hoàn thành tải RaspAP, thì truy cập địa chỉ raspap.local tại browser để đăng nhập vào giao diện quản lý interface của RaspAP, tài khoản mặc định là Username: admin; Password: secret

A screenshot of a computer

Description automatically generated



*Giao diện của RaspAP*

A screenshot of a computer

Description automatically generated

*Cấu hình cơ bản*

A screenshot of a computer

Description automatically generated

*Cài đặt bảo mật*

A screenshot of a computer

Description automatically generated

## CHƯƠNG 4: DEBUG QUÁ TRÌNH HỆ THỐNG

### Mục tiêu của việc Debug

Chương này sẽ tập trung vào việc tìm hiểu quá trình kết nối giữa client và RaspAP dưới mức hệ thống, từ giai đoạn **scanning**, **authentication**, **association**. Việc này giúp nhận diện các vấn đề xảy ra trong các bước thiết lập mạng và cải thiện hiệu suất hệ thống.

### Các công cụ và phương pháp Debug

#### Các công cụ

* + Mã nguồn hostapd
  + Công cụ gỡ lỗi GDB (GNU Debugger)

### Quá trình phát beacon

* 1. Quá trình Scanning trong mạng Wifi
     1. Chủ động (Active Scanning)
* Client gửi các gói tin Probe Request trên một hoặc nhiều kênh (channel) để tìm kiếm AP.
* AP nhận được gói Probe Request và trả lời bằng Probe Response
* Client sử dụng thông tin trong Probe Response để xây dựng danh sách các AP khả dụng (bao gồm SSID, cường độ tín hiệu, loại mã hóa, v.v.)
  + 1. Bị động (Passive Scanning)
* Client "nghe" các gói tin Beacon Frame được phát định kỳ bởi các AP trên các kênh.
* Dựa vào các Beacon Frame, client xây dựng danh sách AP khả dụng.

A diagram of a flowchart

Description automatically generated

* 1. Hàm hostapd\_setup\_interface()

A computer screen with colorful text

Description automatically generated

Cách hoạt động: Hàm này khởi tạo giao diện driver, kiểm tra cấu hình, và thiết lập các tham số driver dựa trên cấu hình. Nó sẽ ngắt kết nối của các thiết bị cũ (flush stations), đặt kênh, mã hóa, tín hiệu beacon và các liên kết WDS dựa trên cấu hình.

* 1. Hàm setup\_interface()

A screenshot of a computer program

Description automatically generated

Cách hoạt động: Hàm thực hiện các bước thiết lập ban đầu cho giao diện trong hostapd

* + Đặt lại trạng thái teardown của giao diện.
  + Lấy tên radio (phy) nếu chưa được xác định.
  + Đồng bộ hóa thông tin driver giữa các giao diện BSS (Basic Service Set).
  + Kiểm tra và xác nhận cấu hình BSSID.
  + Khởi tạo giao diện điều khiển (control interface).
  + Cấu hình mã quốc gia (country code).
  + Nếu cần thiết, đăng ký thao tác cập nhật danh sách kênh (channel list) và đợi kết quả trước khi tiếp tục thiết lập bổ sung (setup\_interface2())
  1. Hàm setup\_interface2()

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer program

Description automatically generated

Hàm tiếp tục thiết lập bổ sung cho giao diện Wi-Fi sau khi các bước cơ bản trong setup\_interface đã hoàn thành. Các bước trong hàm tập trung vào:

* Lấy và cấu hình các tính năng phần cứng của giao diện.
* Kiểm tra và thiết lập chế độ hoạt động (hw\_mode), kênh (channel), băng thông (channel width), và các khả năng Wi-Fi (HE 6Hz, HT (High Throughput), DFS).
* Hoàn thành việc khởi tạo giao diện hoặc đợi callback trong trường hợp cần thêm thời gian để xác định kênh hoặc chế độ hoạt động.
* Gọi hàm để thực hiện các bước cuối cùng (thiết lập beacon, encryption, v.v.) và kích hoạt giao diện.
  1. Hàm hostapd\_setup\_interface\_complete()

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Hàm này hoàn tất quá trình khởi tạo giao diện Wi-Fi. Nó xử lý các bước cuối cùng của việc đồng bộ hóa nhiều giao diện nếu cần và kiểm tra trạng thái khởi tạo.

Hàm có thể được gọi lại từ các callback khác (như khi hoàn tất các bước DFS). Nếu có lỗi xảy ra hoặc tất cả giao diện đã sẵn sàng, nó sẽ thực hiện các bước phù hợp như kích hoạt giao diện hoặc dừng khi có lỗi.

* 1. Hàm hostapd\_setup\_interface\_complete\_sync()

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

Hàm hostapd\_setup\_interface\_complete\_sync() chịu trách nhiệm hoàn thành quá trình thiết lập giao diện trong Hostapd, đảm bảo các thông số kỹ thuật đã được áp dụng đầy đủ và giao diện đã sẵn sàng hoạt động

* 1. Hàm hostapd\_setup\_bss()

A computer screen with text

Description automatically generated

Tóm tắt quy trình chính

1. Khởitạo: Kiểm tra trạng thái, thiết lập địa chỉ MAC và thêm giao diện.
2. Cấu hình bảo mật: Thiết lập SSID, WPA-PSK, và các thành phần bảo mật khác.
3. Khởi tạo dịch vụ bổ sung: ACL, QoS, snooping.
4. Phát sóng: Bắt đầu beacon để thiết bị khác kết nối.
5. Khóa bảo mật: Khởi tạo khóa WPA để hoàn tất bảo mật.
   1. Hàm hostapd\_start\_beacon()

A computer screen shot of a program

Description automatically generated

Hàm hostapd\_start\_beacon() chịu trách nhiệm khởi động beacon trên Access Point (AP). Hàm là bước trung gian quan trọng trong quá trình phát mạng: kiểm tra và gửi beacon để phát sống SSID, đảm bảo không có kết nối cũ và cập nhật trạng hoạt động của driver.

* 1. Hàm ieee802\_11\_set\_beacon()

A computer screen with text

Description automatically generated

Hàm ieee802\_11\_set\_beacon() chịu trách nhiệm gọi hàm thiết lập beacon chính (\_ieee802\_11\_set\_beacon) và kiểm tra xem có cần đồng bộ beacon trên các giao diện khác hay không.

* 1. Hàm \_\_ieee802\_11\_set\_beacon()

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer screen shot of text

Description automatically generated

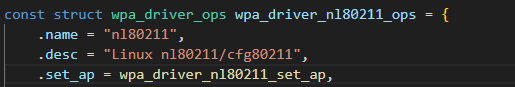
Hàm \_\_ieee802\_11\_set\_beacon chịu trách nhiệm thiết lập và cấu hình các tham số cần thiết để phát sóng beacon, một loại khung dữ liệu quảng bá thông tin của Access Point (AP) trong mạng Wi-Fi. Bao gồm xây dựng cấu trúc tham số AP (wpa\_driver\_ap\_params), xây dựng các gói tin phản hồi, thiết lập thông số tần số, và gửi cấu trúc params đến driver để hiện thực phát sóng beacon

* 1. Hàm hostapd\_drv\_set\_ap()

A screen shot of a computer code

Description automatically generated

Hàm này là một lớp trung gian giữa tầng ứng dụng (hostapd) và tầng driver, giúp chuyển các tham số cấu hình từ ứng dụng xuống phần cứng thông qua driver. Nó đảm bảo rằng chỉ các thiết bị và driver có hỗ trợ chức năng thiết lập AP mới thực hiện thao tác này.



Trong cấu trúc wpa\_driver\_ops, trường .set\_ap là con trỏ hàm được sử dụng để thực hiện cấu hình AP thông qua driver, trước khi call hàm, thì hostapd\_drv\_set\_ap() kiểm tra xem driver đang sử dụng có định nghĩa hàm không rồi call.

* 1. Hàm wpa\_driver\_nl80211\_set\_up()

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

Hàm được sử dụng để thiết lập 1 thông điệp netlink. Thông điệp này là một cơ chế giao tiếp giữa kernel và user space trong Linux, thường được sử dụng để cấu hình các chức năng mạng. Cấu trúc của một Netlink message gồm nhiều phần, trong đó chứa các thông tin cần thiết để truyền tải dữ liệu hoặc thực hiện các yêu cầu từ user space đến kernel và ngược lại.

* 1. Hàm send\_and\_recv()

Hàm này rất quan trọng khi làm việc với giao thức Netlink (đặc biệt trong NL80211), vì nó giúp tự động xử lý nhiều trường hợp phức tạp liên quan đến việc gửi và nhận thông điệp.

Quy trình tổng quát

1. Sao chép callback từ socket Netlink.
2. Cấu hình các tùy chọn hỗ trợ Netlink.
3. Gửi thông điệp đến kernel.
4. Thiết lập các callback để xử lý phản hồi.
5. Vòng lặp chờ phản hồi từ kernel và xử lý lỗi (nếu có).
6. Giải phóng tài nguyên và trả về kết quả.

### Quá trình nhận gói tin và xử lý sự kiện association

A screenshot of a computer screen

Description automatically generated

Sơ đồ trên minh họa quy trình xử lý sự kiện association trong nl80211, từ việc đăng ký lắng nghe các sự kiện từ Netlink socket cho đến xử lý và cập nhật trạng thái kết nối của STA. Các bước cụ thể

Lắng nghe sự kiện

* + Để có thể lắng nghe các sự kiện hệ thống sẽ đăng ký một Netlink socket thông qua hàm eloop\_register\_read\_sock() và theo dõi nó bằng bảng quản lý các socket trong hàm eloop\_sock\_table\_add\_sock().
  + Khi kernel gửi sự kiện Netlink có liên quan đến association thì nó sẽ callback tới hàm wpa\_driver\_nl80211\_event\_receive().

Xử lý sự kiện Netlink

* + wpa\_driver\_nl80211\_event\_receive() nhận dữ liệu từ Netlink socket và xác định loại sự kiện.
  + Với sự kiện liên quan tới association, hàm này gọi hàm nl80211\_new\_station\_event() để xử lý chi tiết.

Phân tích sự kiện STA

* + nl80211\_new\_station\_event() trích xuất thông tin từ sự kiện Netlink (địa chỉ MAC, trạng thái kết nối, v.v.).
  + Sau đó, chuyển tiếp thông tin đến tầng cao hơn thông qua hàm drv\_event\_assoc().

Xử lý logic tại wpa\_supplicant

* + drv\_event\_assoc() gọi wpa\_supplicant\_event**()** và truyền thông tin về STA, bao gồm địa chỉ MAC và trạng thái kết nối.
  + Tại hàm wpa\_supplicant\_event() thực hiện các bước kiểm tra cấu hình và xác minh xem STA có nằm trong danh sách được kết nối hay không. Nếu STA hợp lệ thì sự kiện association được chuyển tiếp đến hostapd để tiếp tục xử lý.

Cập nhật trạng thái tại hostapd

* + hostapd\_notif\_assoc() nhận sự kiện từ wpa\_supplicant và cập nhật trạng thái STA, thêm STA vào danh sách quản lý. Sau đó gọi hàm hostapd\_new\_assoc\_sta() để thiết lập thông số kết nối và chuẩn bị các bước bảo mật.

Khởi động quy trình bảo mật

* + Cuối cùng, hàm wpa\_auth\_sta\_associated() được gọi để bắt đầu quy trình 4-way handshake, đảm bảo rằng STA và AP thiết lập được khóa mã hóa chung để giao tiếp an toàn.

Một số hàm quan trọng

A screen shot of a computer program

Description automatically generated

Hàm này được gọi khi có sự kiện mới từ Netlink socket. Mục đích của nó là nhận và xử lý các thông điệp sự kiện, như sự kiện association từ nl80211.

Hàm này nhận các thông điệp từ Netlink qua hàm nl\_recvmsgs() và kiểm tra xem các thông điệp có bị lỗi hay không nếu có sẽ thông báo cho người dùng.A screen shot of a computer program

Description automatically generated

Hàm nl80211\_new\_station\_event() xử lý sự kiện association của một STA mới (một thiết bị mới kết nối vào AP). Hàm này lấy thông tin từ các thuộc tính Netlink và quyết định các bước tiếp theo trong quá trình xử lý sự kiện.

Đầu tiên nó sẽ kiểm tra có địa chỉ MAC hay không nếu không có hàm sẽ kết thúc.

Sau đó sẽ kiểm tra kiểu giao diện có phải là AP và device\_ap\_sme có được kích hoạt hay không

Tiếp tục kiểm tra xem có thông tin về MLO hay không. Sau khi lấy được thông tin MLO thì nó sẽ lấy các thông tin Assoc Req Ies, Assoc Resp Ies và in ra dưới dạng hex dump

Cuối cùng nó sẽ chuyển các thông tin tới hàm drv\_event\_assoc() để tiếp tục xử lý các sự kiện association

A screen shot of a computer program

Description automatically generated  
Hàm này xử lý các sự kiện khác nhau từ wpa\_supplicant (phần mềm quản lý kết nối mạng Wi-Fi). Cụ thể, nó kiểm tra từng loại sự kiện và thực hiện các hành động tương ứng.

Khởi tạo và Debugging:

* Đoạn mã kiểm tra xem có bật chế độ debug không. Nếu có, nó sẽ kiểm tra loại sự kiện, và nếu đó là một sự kiện nhận gói tin quản lý (management frame) như beacon hay yêu cầu probe, mức độ logging sẽ được nâng lên (MSG\_EXCESSIVE). Sau đó, thông tin sự kiện sẽ được ghi lại vào log.

Các trường hợp sự kiện (switch)

* **EVENT\_MICHAEL\_MIC\_FAILURE:** Xử lý sự kiện lỗi MIC (Message Integrity Check) trong WPA.
* **EVENT\_SCAN\_RESULTS:** Xử lý kết quả quét (scan) và gọi callback nếu có.
* **EVENT\_WPS\_BUTTON\_PUSHED:** Xử lý khi nút WPS (Wi-Fi Protected Setup) bị nhấn.
* **EVENT\_TX\_STATUS:** Xử lý trạng thái truyền (transmission status) cho các gói tin quản lý hoặc dữ liệu.
* **EVENT\_EAPOL\_TX\_STATUS:** Xử lý trạng thái gửi gói tin EAPOL (EAP over LAN).
* **EVENT\_DRIVER\_CLIENT\_POLL\_OK:** Xử lý sự kiện khi kiểm tra trạng thái của client.
* **EVENT\_RX\_FROM\_UNKNOWN:** Xử lý khi nhận gói tin từ một STA không xác định.
* **EVENT\_RX\_MGMT:** Xử lý sự kiện nhận gói tin quản lý.
* **EVENT\_RX\_PROBE\_REQ:** Xử lý yêu cầu probe.
* **EVENT\_NEW\_STA:** Xử lý sự kiện STA mới kết nối.
* **EVENT\_EAPOL\_RX:** Xử lý sự kiện nhận gói tin EAPOL.
* **EVENT\_ASSOC:** Xử lý sự kiện kết nối (association).
* **EVENT\_PORT\_AUTHORIZED:** Xử lý khi một cổng được ủy quyền cho một STA.
* **EVENT\_UPDATE\_DH:** Cập nhật thông tin DH (Diffie-Hellman) cho một STA.
* **EVENT\_DISASSOC và EVENT\_DEAUTH:** Xử lý khi một STA bị ngắt kết nối hoặc bị từ chối xác thực.
* **EVENT\_STATION\_LOW\_ACK:** Xử lý sự kiện khi STA có ACK thấp.
* **EVENT\_AUTH:** Xử lý sự kiện xác thực (authentication).
* **EVENT\_CH\_SWITCH:** Xử lý sự kiện chuyển kênh (channel switch).
* **EVENT\_CONNECT\_FAILED\_REASON:** Xử lý lý do kết nối không thành công.
* **EVENT\_SURVEY:** Xử lý sự kiện khảo sát kênh.
* **EVENT\_DFS\_RADAR\_DETECTED:** Xử lý sự kiện radar DFS (Dynamic Frequency Selection) được phát hiện.
* **EVENT\_INTERFACE\_ENABLED và EVENT\_INTERFACE\_DISABLED:** Xử lý sự kiện bật/tắt giao diện mạng.
* **EVENT\_ACS\_CHANNEL\_SELECTED:** Xử lý khi kênh ACS (Automatic Channel Selection) được chọn.
* **EVENT\_STATION\_OPERATING\_MODE\_CHANGED:** Xử lý khi chế độ hoạt động của STA thay đổi.

### Quá trình xác thực

Authentication (xác thực) là một bước quan trọng trong quá trình kết nối giữa client (thiết bị người dùng) và Access Point (AP). Mục đích của bước này là đảm bảo rằng client có quyền truy cập vào mạng và thiết lập một kênh bảo mật cho việc truyền thông tiếp theo. Quá trình xác thực giúp ngăn ngừa các kết nối trái phép và bảo vệ mạng khỏi các mối đe dọa từ bên ngoài.

#### **4.5.1 Các Giao thức bảo mật Wi-Fi và Cách Thức Hoạt Động**

Các giao thức bảo mật trong mạng Wi-Fi chủ yếu được sử dụng để bảo vệ mạng khỏi các truy cập trái phép và đảm bảo an toàn cho dữ liệu truyền qua mạng. Dưới đây là chi tiết về các phương pháp bảo mật Wi-Fi phổ biến hiện nay:

**WEP (Wired Equivalent Privacy)**

* **Cách hoạt động**: WEP là giao thức bảo mật đầu tiên được sử dụng cho các mạng không dây, sử dụng mã hóa RC4 với một khóa có độ dài 40 bit hoặc 104 bit (WEP+ có thể sử dụng 128 bit). Mỗi gói tin trong mạng sẽ được mã hóa bằng khóa này, đảm bảo bảo vệ dữ liệu khi truyền tải qua sóng radio.
* **Nhược điểm**: Mặc dù WEP được thiết kế để cung cấp bảo mật tương đương với các kết nối có dây, nhưng giao thức này có nhiều điểm yếu nghiêm trọng. Các tấn công như tấn công từ điển và tấn công lặp lại có thể dễ dàng bẻ khóa WEP. Chính vì thế, WEP ngày nay không còn an toàn và ít được sử dụng.

**WPA (Wi-Fi Protected Access)**

* **Cách hoạt động**: WPA được phát triển như là một sự thay thế tạm thời cho WEP nhằm khắc phục các vấn đề bảo mật của WEP. WPA sử dụng TKIP (Temporal Key Integrity Protocol) để thay thế mã hóa RC4 trong WEP. TKIP có khả năng thay đổi khóa mã hóa mỗi khi một gói tin mới được truyền đi, giúp nâng cao bảo mật. WPA cũng sử dụng một cơ chế để kiểm tra tính toàn vẹn của dữ liệu, giúp ngăn ngừa tấn công giả mạo và nghe lén.
* **Ưu điểm**: WPA cải thiện bảo mật đáng kể so với WEP và kiểm tra tính toàn vẹn của dữ liệu trong mỗi gói tin. Tuy nhiên, WPA vẫn sử dụng TKIP, vốn không mạnh mẽ và an toàn như các tiêu chuẩn bảo mật mới hơn.
* **Nhược điểm**: Dù WPA cải thiện bảo mật, nhưng phương thức TKIP vẫn có thể bị tấn công, ví dụ như qua phương pháp tấn công phá khóa. Do đó, WPA không được coi là đủ mạnh mẽ cho các mạng yêu cầu bảo mật cao.

**WPA2 (Wi-Fi Protected Access 2)**

* **Cách hoạt động**: WPA2 là phiên bản cải tiến của WPA, sử dụng mã hóa AES (Advanced Encryption Standard) thay cho TKIP. AES là một thuật toán mã hóa mạnh mẽ và an toàn hơn nhiều so với RC4 và TKIP. WPA2 hỗ trợ hai phương thức bảo mật chính:
  + **WPA2-Personal (WPA2-PSK)**: Sử dụng mật khẩu chung cho tất cả các thiết bị trong mạng.
  + **WPA2-Enterprise**: Sử dụng hệ thống xác thực 802.1X với máy chủ RADIUS để quản lý người dùng và cấp phát mật khẩu cho từng thiết bị.
* **Ưu điểm**: WPA2 với mã hóa AES cung cấp một mức độ bảo mật rất cao, giúp ngăn ngừa các tấn công giả mạo và nghe lén. AES được coi là một trong những thuật toán mã hóa mạnh mẽ và an toàn nhất hiện nay. WPA2 là chuẩn bảo mật tối thiểu cho hầu hết các mạng Wi-Fi hiện nay và được sử dụng rộng rãi trong các mạng gia đình và doanh nghiệp.
* **Nhược điểm**: Dù WPA2 rất an toàn, nó vẫn có thể bị ảnh hưởng bởi các lỗi cấu hình hoặc mật khẩu yếu. Nếu thiết bị hoặc mạng không hỗ trợ AES hoặc không được cấu hình đúng, bảo mật của WPA2 có thể bị giảm sút.

**Tóm lại:**

Các giao thức bảo mật Wi-Fi như WEP, WPA, và WPA2 có các mức độ bảo vệ khác nhau, và được sử dụng tùy theo yêu cầu bảo mật của mạng. WEP là giao thức cũ và không an toàn, trong khi WPA cải thiện bảo mật nhưng vẫn có những điểm yếu. WPA2, với mã hóa AES, giao thức bảo mật mạnh mẽ và an toàn nhất hiện nay và được khuyến nghị sử dụng cho tất cả các mạng Wi-Fi.

Trong báo cáo này quá trình sẽ tập trung vào giao thức WPA2-Personal.

#### **4.5.2 Quá trình bắt tay 4 bước trong xác thực**

Quá trình bắt tay 4 bước (4-Way Handshake) là một phần quan trọng trong cơ chế xác thực giữa client (thiết bị người dùng) và Access Point (AP) trong mạng Wi-Fi, đặc biệt là trong giao thức WPA2. Quá trình này đảm bảo rằng cả hai bên (client và AP) có thể xác thực lẫn nhau và thiết lập các khóa bảo mật cần thiết để truyền thông an toàn.

**Mục Đích Của Quá Trình Bắt Tay 4 bước:**

* Xác thực thiết bị: Đảm bảo rằng chỉ những thiết bị hợp lệ mới có thể kết nối với mạng.
* Thiết lập khóa bảo mật: Tạo ra các khóa tạm thời (PTK và GTK) để mã hóa lưu lượng giữa client và AP.
* Ngăn ngừa tấn công: Đảm bảo tính toàn vẹn và bảo mật của thông điệp trong quá trình xác thực.

A diagram of a computer system

Description automatically generated

*Bắt tay 4 bước*

**Thông điệp 1 (WAP → Client):**

WAP gửi thông điệp EAPOL chứa giá trị nonce (ANonce) cho client. Client sử dụng ANonce để tính toán Pairwise Transient Key (PTK):

PTK = PRF (PMK + ANonce + SNonce + MAC(AA) + MAC(SA))

**Thông điệp 2 (Client → WAP):**

Client gửi lại thông điệp EAPOL với SNonce (nonce do client tạo) và MIC (Message Integrity Code) để kiểm tra tính toàn vẹn của thông điệp.

**Thông điệp 3 (WAP → Client):**

WAP xác minh thông điệp từ client và gửi Group Temporal Key (GTK) cùng MIC khác. GTK bảo vệ các gói tin nhóm (multicast/broadcast).

**Thông điệp 4 (Client → WAP):**

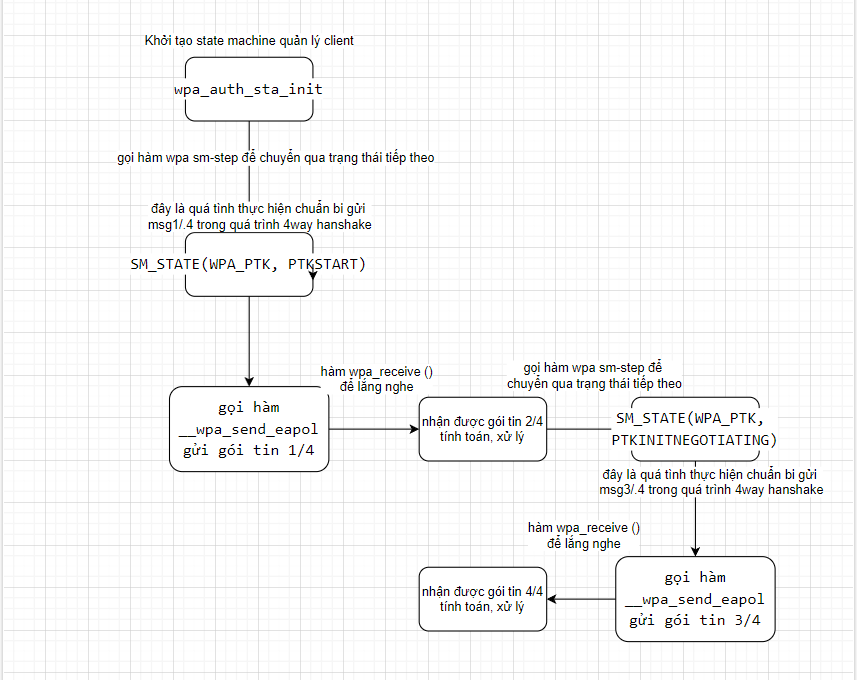
Client gửi thông điệp cuối cùng xác nhận rằng PTK và GTK đã được cài đặt thành công, cho phép mã hóa lưu lượng giữa client và WAP.

**Các từ khoá thường dùng trong quá trình này**

* PMK (Pairwise Master Key): Khóa gốc dùng để tạo ra PTK, thường được tạo từ PSK (Pre-shared Key) trong WPA2-PSK.
* PTK (Pairwise Transient Key): Khóa tạm thời được tạo ra từ ANonce và SNonce, sử dụng để mã hóa và giải mã dữ liệu giữa client và WAP.
* GTK (Group Temporal Key): Khóa bảo vệ các gói tin nhóm, cho phép WAP mã hóa thông điệp gửi đến nhiều client cùng lúc.
* MIC (Message Integrity Code): Mã kiểm tra tính toàn vẹn của thông điệp, đảm bảo rằng không có sự giả mạo hoặc thay đổi nội dung.
* State Machine: Cấu trúc quản lý các trạng thái trong quá trình thiết lập khóa bảo mật, đảm bảo các bước xác thực diễn ra đúng thứ tự.
* KDE (Key Data Element): Các thành phần thông tin liên quan đến khóa, bao gồm thông tin bảo mật được trao đổi giữa client và AP.
* EAPOL (Extensible Authentication Protocol over LAN): Giao thức dùng để trao đổi thông điệp xác thực và khóa giữa client và AP, là phần quan trọng trong quá trình handshake bảo mật trong Wi-Fi.
* wpa\_group là một cấu trúc trong mã nguồn, được sử dụng để quản lý các thông tin liên quan đến Group Key như GTK, Key Id, Group Cipher, GTK Re-Keying, Group Key Security

#### **4.5.3 Phân tích quá trình debug trong file wpa\_auth.c**

File wpa\_auth.c đóng vai trò quan trọng trong việc xử lý quá trình xác thực của Access Point trong mạng Wi-Fi. Đây là nơi diễn ra các hoạt động xác thực giữa AP và client, bao gồm việc nhận và xử lý các thông điệp EAPOL, xác minh tính toàn vẹn của thông điệp, và tạo ra các khóa bảo mật. Việc debug trong file này không chỉ giúp phát hiện lỗi mà còn cung cấp cái nhìn sâu sắc về cách thức hoạt động của hệ thống xác thực, từ đó cải thiện hiệu suất và độ tin cậy của mạng.



Các hàm

**wpa\_auth\_state\_init**

Hàm wpa\_auth\_sta\_init() khởi tạo state machine quản lý trạng thái WPA cho client, lưu địa chỉ MAC, liên kết với cấu hình Access Point và Group Key, đồng thời trả về con trỏ đến cấu trúc này. Hàm thường được gọi khi một STA gửi yêu cầu kết nối tới AP.

A screenshot of a computer program

Description automatically generated

**wpa\_sm\_step()**

Hàm wpa\_sm\_step() điều khiển tiến trình của state machine cho WPA, xử lý các bước của state machine WPA (như WPA\_PTK, WPA\_PTK\_GROUP) đảm bảo rằng quá trình xác thực và quản lý khóa WPA được thực hiện đúng cách. Quá trình chuyển trạng thái của state machine được thực hiện thông qua macro SM\_STEP\_RUN(WPA\_PTK). Macro này thực hiện việc gọi các hàm tương ứng với từng trạng thái trong nhóm state machine WPA\_PTK.

A computer screen shot of text

Description automatically generated

A screen shot of a computer program

Description automatically generated

**\_\_wpa\_send\_eapol()**

Hàm \_\_wpa\_send\_eapol() được sử dụng để gửi một thông điệp EAPOL-Key frame từ Access Point đến Client.

A screen shot of a computer program

Description automatically generated

Trích xuất cấu hình: Lấy thông tin từ wpa\_auth để xác định các tham số như thuật toán mã hóa, phiên bản EAPOL-Key, và cặp khóa (pairwise).

Tính toán độ dài: Xác định kích thước MIC, header EAPOL-Key, và dữ liệu mã hóa.

Xác định phiên bản và cờ (flags): Chọn phiên bản EAPOL-Key và thiết lập các cờ như secure, MIC, install, dựa trên cấu hình và trạng thái hiện tại.

A screen shot of a computer program

Description automatically generated

* Tính toán kích thước: Đoạn mã điều chỉnh chiều dài dữ liệu khóa (key\_data\_len) để phù hợp với yêu cầu mã hóa (nếu có).
* Cấp phát bộ nhớ: Sử dụng os\_zalloc để tạo không gian lưu trữ cho khung EAPOL-Key.
* Khởi tạo header: Thiết lập các trường cơ bản (phiên bản, loại khung, độ dài) trong phần header của EAPOL-Key.

A computer screen with colorful text

Description automatically generated

* Khởi tạo con trỏ: Trỏ đến các vùng nhớ của EAPOL-Key (header, MIC, và dữ liệu khóa).
* Thiết lập thông tin khung: Xác định loại EAPOL-Key, phiên bản, và các thông số mã hóa (key type, key info, key length).

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

Đoạn mã này xử lý việc mã hóa dữ liệu trong KDE (Key Data Element) khi cần bảo mật dữ liệu trước khi gửi qua khung EAPOL-Key.:

* Khởi tạo bộ đệm buf để lưu trữ KDE và thêm padding nếu cần.
* Sao chép KDE vào buf và thêm padding.
* Ghi log dữ liệu plaintext (chưa mã hóa) để hỗ trợ debug.
* Mã hóa dữ liệu KDE bằng AES-WRAP với khóa KEK.
* Cập nhật chiều dài của dữ liệu mã hóa vào trường thông tin MIC.

A screen shot of a computer program

Description automatically generated

* Đảm bảo rằng MIC (Message Integrity Code) được tính toán chính xác cho khung EAPOL-Key nếu yêu cầu.
* Kiểm tra tính hợp lệ của PTK trước khi gửi khung EAPOL-Key, đảm bảo không có lỗi khi truyền tải dữ liệu bảo mật.

A computer code on a black background

Description automatically generated

* Cập nhật số lượng khung EAPOL đã gửi: Hàm wpa\_auth\_set\_eapol giúp theo dõi quá trình gửi khung EAPOL, đảm bảo rằng mỗi lần gửi một khung EAPOL, số lượng khung đã được gửi sẽ được tăng lên.
* Gửi khung EAPOL-Key: Hàm wpa\_auth\_send\_eapol thực hiện việc gửi khung EAPOL-Key đến thiết bị xác thực (sm->addr).

**Sending 1/4 msg**

Thông điệp 1/4 trong quá trình bắt tay 4 bước được gửi khi trạng thái của state machine chuyển sang PTKSTART. Đây là bước đầu tiên trong quá trình xác thực và thiết lập kết nối bảo mật giữa client và Access Point.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

Khi trạng thái là state machine là PTKSTART, quá trình 4-Way Handshake bắt đầu, AP sẽ gửi EAPOL-Key Message 1/4 đến client, chứa thông tin về nonce (ngẫu nhiên), replay counter và thông tin khóa dùng trong quá trình xác thực.

**Sending 3/4 msg**

Thông điệp 3/4 trong quá trình bắt tay 4 bước được gửi khi Access Point xác minh thành công thông điệp 2/4 từ client và state machine chuyển sang trạng thái PTKINITNEGOTIATING. Tiến hành chuẩn bị khoá GTK gửi cho client.

A computer screen shot of a program

Description automatically generated

Khi trạng thái là PTKINITNEGOTIATING, AP khởi tạo dữ liệu cần thiết cho gói tin 3/4 của 4-Way Handshake (bao gồm RSC, KDE, GTK, WPA IE), kiểm tra số lần gửi lại để ngăn Replay Attack và đảm bảo an toàn bằng cách ngắt kết nối nếu vượt quá giới hạn thử.

A computer screen shot of a program

Description automatically generated

Đoạn mã trên thực hiện các bước thiết lập và xác thực khóa cho quá trình 4-Way Handshake khi sử dụng WPA2. Cụ thể, mã thực hiện ghi log thông báo về quá trình gửi gói tin EAPOL 3/4, sau đó cài đặt khóa mã hóa cho kết nối WPA2 và xử lý các trường hợp lỗi, ngắt kết nối nếu không thành công trong việc cài đặt khóa.

A screenshot of a computer program

Description automatically generated

Sau khi thiết lập mã hoá cho kết nối , bật secure lên giá trị 1.

* Lấy Group Temporal Key (GTK): Truy xuất giá trị GTK từ nhóm theo Group Nonce (GN), dùng để mã hóa và giải mã dữ liệu trong mạng.
* Tạo GTK ngẫu nhiên khi cần thiết: Nếu cần, mã sẽ tạo ra một GTK ngẫu nhiên thay vì sử dụng GTK cũ, giúp tránh việc tái sử dụng khóa nhóm, đảm bảo an toàn cho kết nối.
* Cập nhật thông tin và chuẩn bị mã hóa: Cập nhật chỉ số GTK, giá trị RSC và bật trạng thái mã hóa (encr = 1) để bảo vệ dữ liệu trong quá trình truyền nhận.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Đoạn mã tính toán độ dài thông điệp KDE, cấp phát bộ nhớ cho KDE, sao chép WPA IE vào bộ nhớ, thêm các KDEs như Extended Key ID, GTK, IEEE 802.11w, OCV, gửi thông điệp EAPOL với các cờ bảo mật và cuối cùng giải phóng bộ nhớ đã cấp phát cho KDE và các bộ đệm WPA IE.

**wpa\_receive()**

Hàm wpa\_receive thực hiện các kiểm tra và xử lý các lỗi liên quan đến tính hợp lệ của các thông điệp EAPOL-Key trong quá trình 4-way handshake. Các bước này giúp đảm bảo rằng các thông điệp nhận được là hợp lệ, an toàn và không bị gian lận, đồng thời duy trì sự bảo mật trong giao tiếp giữa client và access point.

A screenshot of a computer program

Description automatically generated

Đoạn mã trong hàm wpa\_receive kiểm tra tính hợp lệ của phiên bản mã hóa và tính toàn vẹn trong gói EAPOL-Key. Nó xử lý các thông điệp REQUEST, PAIRWISE\_2, PAIRWISE\_4, và GROUP\_2, trích xuất phiên bản key\_info để xác định phương pháp mã hóa.

* AES-128-CMAC: Báo lỗi nếu không dùng đúng.
* HMAC-SHA1-AES: Báo lỗi nếu không dùng đúng phiên bản.
* AKM-defined: Báo lỗi nếu không tuân thủ phiên bản 0.

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Đoạn mã trên xử lý các thông điệp EAPOL-Key, kiểm tra trạng thái của máy chủ sm và quyết định hành động cần thực hiện cho từng loại thông điệp (PAIRWISE\_2, PAIRWISE\_4, GROUP\_2, REQUEST).

A screen shot of a computer program

Description automatically generated

* Kiểm tra cờ ACK: Đảm bảo rằng thông điệp EAPOL không có cờ ACK khi không cần thiết.
* Kiểm tra cờ MIC: Đảm bảo rằng thông điệp có cờ MIC nếu không sử dụng FILS, nhằm bảo vệ tính toàn vẹn của thông điệp.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Đoạn mã này xử lý thông điệp EAPOL-Key trong quá trình 4-way handshake, bao gồm việc xác thực MIC, xử lý các yêu cầu từ client, quản lý các trạng thái liên quan đến PTK, GTK, và xử lý các lỗi có thể xảy ra.

A computer code on a black background

Description automatically generated

Giải phóng bộ nhớ cũ và lưu thông điệp EAPOL-Key vừa nhận. Phân tích và lưu các thông tin chính từ thông điệp, bao gồm:

* Loại khóa (*pairwise* hoặc *group key*).
* Trạng thái bảo mật và yêu cầu.
* Giá trị nonce.

Kích hoạt state machine để xử lý các bước tiếp theo trong quá trình thiết lập hoặc xác thực khóa WPA/WPA2.

## CHƯƠNG 5: KẾT LUẬN

Trong đồ án này, nhóm em đã cài đặt thành công RaspAP trên một Raspberry Pi 4, biến nó thành một Access Point để các thiết bị có thể kết nối. Sau đó, nhóm đã tiến hành debug quá trình kết nối của người dùng với Access Point, đặc biệt tập trung vào các hàm xử lý trong quá trình probing, authentication và association. Chúng em đã phân tích các bước này để hiểu rõ hơn cách thức hoạt động của RaspAP trong môi trường thực tế và tìm ra các vấn đề tiềm ẩn trong việc xử lý kết nối.

Nhóm chúng em sẽ tiếp tục nghiên cứu sâu hơn để hiểu rõ hơn chức năng cụ thể của từng hàm trong quá trình kết nối, đồng thời cải tiến hiệu suất của hệ thống. Bên cạnh đó, nhóm sẽ thử nghiệm và mở rộng thêm các tính năng bảo mật, như hỗ trợ WPA3, để tăng cường độ bảo mật của Access Point.

## TÀI LIỆU THAM KHẢO

[1]R. Rosen, “Linux Wireless - Linux Kernel Networking (4)- advanced topics,” 2009. Available: <http://www.haifux.org/lectures/206/wirelessLec.pdf>

**[2]** Richard, “Linux kernel WiFi stack basics,” *Substack.com*, Mar. 07, 2024. <https://wifidiving.substack.com/p/linux-kernel-wifi-stack-basics> (accessed Dec. 01, 2024).

‌**[3]** admin, “4-Way Handshake,” *WiFi*, Jan. 24, 2019. <https://www.wifi-professionals.com/2019/01/4-way-handshake>

[4]“RaspAP Documentation,” *Raspap.com*, 2024. <https://docs.raspap.com/> (accessed Dec. 01, 2024).

[5]latelee, “GitHub - latelee/hostapd: hostapd from git://w1.fi/srv/git/hostap.git,” *GitHub*, 2016. <https://github.com/latelee/hostapd> (accessed Dec. 01, 2024).

‌

‌

‌