TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG
MÔN HỆ THỐNG NHÚNG MẠNG KHÔNG DÂY

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Tên đồ án: Cấu hình RaspAP và xây dựng ứng dụng "Network Security Monitor" trên Raspberry Pi 4

Nhóm thực hiện:

stt	Họ và tên	Mã số sinh viên
1	Lương Cao Thắng	22521328
2	Nguyễn Hữu Thắng	22521340
3	Phạm Xuân Tuấn Anh	22520071

Giảng viên hướng dẫn: PGS.TS LÊ TRUNG QUÂN

Thời gian thực hiện: 10/10/2024 – 17/11/2024

Thành phố Hồ Chí Minh, ngày 17 tháng 11 năm 2024

TÓM TẮT

Tên đồ án: Xây dựng ứng dụng "Network Security Monitor" trên Raspberry Pi 4.
Đồ án "Network Security Monitor" trên Raspberry Pi 4 hướng đến phát triển một ứng dụng giám sát bảo mật mạng nội bộ với khả năng bắt và phân tích gói tin, phát hiện hành vi bất thường như quét cổng hoặc tấn công DoS. Đồng thời cung cấp một giao diện web sử dụng Flask để người dùng xem báo cáo và cấu hình. Dữ liệu mạng được lưu trữ trong SQLite để phân tích xu hướng và tạo báo cáo, với tối ưu hóa để ứng dụng hoạt động hiệu quả trên phần cứng hạn chế của Raspberry Pi 4.

MỤC LỤC

I. Giới Thiệu:

1.1. Lý do chọn đề tài:

- Trong thời đại công nghệ số, sự phát triển nhanh chóng của các thiết bị kết nối Internet và mạng máy tính đã dẫn đến sự gia tăng không ngừng của các mối đe dọa an ninh mạng. Việc bảo vệ hệ thống mạng khỏi các cuộc tấn công và phát hiện kịp thời các hoạt động bất thường trở thành yếu tố sống còn cho cá nhân và doanh nghiệp. Tuy nhiên, các giải pháp bảo mật mạng thương mại thường đòi hỏi chi phí cao và khó triển khai với các cá nhân hoặc tổ chức nhỏ.

- Raspberry Pi, với khả năng tùy biến cao và chi phí thấp, là lựa chọn lý tưởng cho việc phát triển một giải pháp giám sát và bảo mật mạng nhỏ gọn và hiệu quả. Bằng cách tận dụng sức mạnh của Raspberry Pi 4 Model B kết hợp với các công cụ và thư viện mã nguồn mở như Python, Flask, và Scapy, chúng ta có thể xây dựng một hệ thống giám sát mạng cơ bản, giúp phát hiện và cảnh báo các hoạt động bất thường trên hệ thống mạng nội bộ.

1.2. Mục tiêu của đồ án:

Mục tiêu chính của đồ án này là phát triển một ứng dụng giám sát và bảo mật mạng hoạt động trên Raspberry Pi 4 Model B với các chức năng:

- Giám sát lưu lượng mạng và thu thập các thông tin cơ bản của các gói tin như địa chỉ IP nguồn, địa chỉ IP đích, loại giao thức.
- Phát hiện và nhận diện các hoạt động bất thường, bao gồm các cuộc tấn công mạng phổ biến như quét cổng, tấn công DoS.
- Cung cấp giao diện web thân thiện cho người dùng, hiển thị thông tin tổng quan, các thông báo, và cho phép thiết lập cấu hình.
- Lưu trữ và phân tích log để giúp người dùng có thể theo dõi xu hướng hoạt động mạng trong thời gian dài.

1.3. Phạm vi ứng dụng:

Ứng dụng này được thiết kế để phù hợp với quy mô mạng nhỏ như mạng gia đình, văn phòng nhỏ hoặc tổ chức quy mô nhỏ, nơi không yêu cầu hệ thống bảo mật phức tạp và chi phí cao. Với phần cứng hạn chế của Raspberry Pi, ứng dụng sẽ tập trung vào những tính năng cơ bản nhất nhằm đảm bảo hiệu suất mà vẫn đáp ứng yêu cầu giám sát và bảo mật mạng.

1.4. Phương pháp và công cụ phát triển:

Dự án sẽ được phát triển dựa trên các công cụ và thư viện mã nguồn mở, nhằm đảm bảo tính khả dụng và dễ dàng triển khai cho người dùng phổ thông. Các công cụ và thư viện chính bao gồm:

- **Python**: Ngôn ngữ lập trình chính để xây dựng ứng dụng.
- **Scapy**: Thư viện dùng để bắt và phân tích gói tin mạng.
- **Flask**: Khung phát triển ứng dụng web để tạo giao diện hiển thị và cấu hình.
- **SQLite**: Hệ quản trị cơ sở dữ liệu nhẹ để lưu trữ log và phân tích dữ liệu.

II. Tổng quan về Raspberry Pi và Network Security Monitor

2.1. Giới thiệu về Raspberry Pi 4 Model B:

Raspberry Pi 4 Model B là một trong những phiên bản mới nhất của dòng máy tính mini Raspberry Pi, nổi bật với cấu hình mạnh mẽ và kích thước nhỏ gọn. Raspberry Pi 4 Model B sở hữu CPU ARM Cortex-A72 lõi tứ, tốc độ 1.5GHz, hỗ trợ dung lượng RAM lên đến 4GB, cùng với các cổng kết nối đa dạng như cổng Ethernet, USB 3.0 và cổng HDMI kép. Với những tính năng này, Raspberry Pi 4 có khả năng đáp ứng nhu cầu xử lý của các ứng dụng giám sát và bảo mật mạng quy mô nhỏ, đồng thời tiêu thụ năng lượng thấp.

Các ưu điểm chính của Raspberry Pi 4 Model B trong dự án bao gồm:

- Giá thành hợp lý: Với chi phí thấp, Raspberry Pi là giải pháp tiết kiệm cho các dự án cá nhân và doanh nghiệp nhỏ.
- Tính linh hoạt**:** Raspberry Pi có thể dễ dàng cài đặt các hệ điều hành dựa trên Linux, cho phép tùy biến theo nhu cầu của người dùng.
- Khả năng mở rộng: Hỗ trợ nhiều thư viện và công cụ mã nguồn mở, giúp phát triển các ứng dụng giám sát mạng hiệu quả và tiết kiệm.

2.2. Các vấn đề bảo mật mạng thường gặp:

Trong bối cảnh mạng hiện đại, các cuộc tấn công và mối đe dọa mạng ngày càng đa dạng và tinh vi. Một số vấn đề bảo mật mạng phổ biến bao gồm:

- Tấn công DoS (Denial of Service): Các kẻ tấn công gửi lượng lớn yêu cầu đến máy chủ hoặc thiết bị mạng nhằm làm quá tải hệ thống và khiến nó không thể cung cấp dịch vụ cho người dùng hợp lệ.
- Quét cổng (Port Scanning): Các cuộc tấn công này thường nhằm xác định các cổng mở và dịch vụ đang chạy trên hệ thống để khai thác các lỗ hổng bảo mật.

Các vấn đề này đòi hỏi một hệ thống giám sát và phát hiện bất thường để cảnh báo người dùng kịp thời, bảo vệ tính bảo mật và tính toàn vẹn của hệ thống mạng.

2.3. Tổng quan về Network Security Monitoring (NSM):

Network Security Monitoring (NSM) là một phương pháp tiếp cận an ninh mạng tập trung vào việc thu thập, phân tích và lưu trữ dữ liệu lưu lượng mạng nhằm phát hiện và phản ứng kịp thời với các mối đe dọa. Thay vì chỉ ngăn chặn các cuộc tấn công, NSM cung cấp thông tin chi tiết về hoạt động mạng, giúp người dùng hiểu rõ hơn về các sự kiện và hành vi bất thường trong mạng. Các chức năng chính của một hệ thống NSM bao gồm:

- Thu thập lưu lượng mạng: Bắt và ghi nhận thông tin chi tiết về các gói tin mạng để phân tích và lưu trữ.
- Phân tích và phát hiện các hoạt động bất thường: Áp dụng các thuật toán để nhận diện các mẫu hành vi đáng ngờ như lưu lượng tăng đột biến, các nỗ lực quét cổng hoặc DoS.
- Giao diện giám sát: Cung cấp giao diện người dùng thân thiện để theo dõi tình trạng mạng, xem các cảnh báo và quản lý cấu hình hệ thống.

2.4. Các công nghệ sử dụng trong dự án:

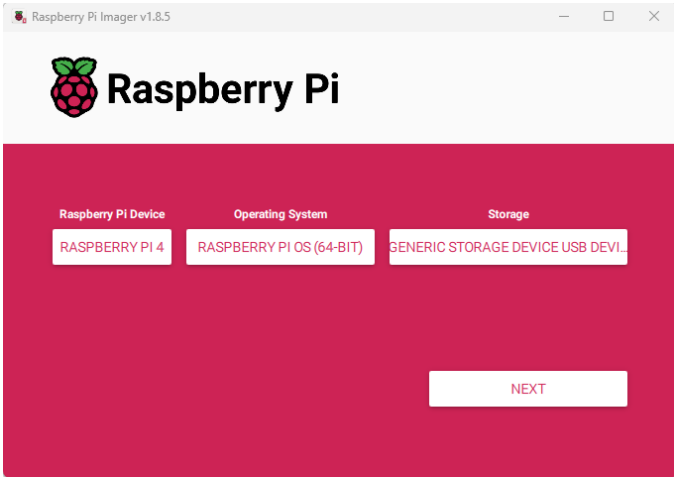
Để xây dựng ứng dụng Network Security Monitor trên Raspberry Pi, các công nghệ và công cụ chính sau đây sẽ được sử dụng:

- Python**:** Là ngôn ngữ chính của dự án nhờ vào khả năng hỗ trợ nhiều thư viện mã nguồn mở, đơn giản và dễ học, phù hợp với phát triển ứng dụng trên Raspberry Pi.
- Scapy: Thư viện Python mạnh mẽ dùng để phân tích lưu lượng mạng, bắt và xử lý gói tin. Scapy cho phép ghi lại các thông tin quan trọng từ gói tin như IP nguồn, IP đích, giao thức, và các flag TCP, phù hợp để phân tích các hoạt động bất thường.
- Flask**:** Một framework Python nhẹ, được sử dụng để xây dựng ứng dụng web phục vụ giao diện hiển thị và quản lý. Flask cho phép triển khai web server trên Raspberry Pi và cung cấp API để người dùng cấu hình và xem báo cáo.
- SQLite**:** Là hệ quản trị cơ sở dữ liệu nhẹ, SQLite phù hợp cho việc lưu trữ log và dữ liệu phân tích trong các hệ thống nhỏ gọn như Raspberry Pi. SQLite giúp lưu trữ thông tin về các sự kiện mạng, cảnh báo, và cung cấp truy vấn để phân tích dữ liệu trong thời gian dài.

III. Thiết lập môi trường phát triển:

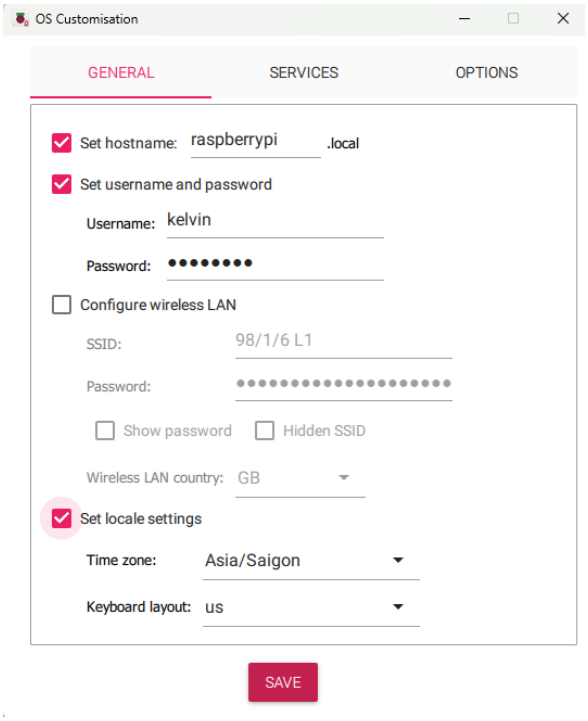
3.1. Cài đặt hệ điều hành Raspbian OS trên Raspberry Pi 4:

- Lựa chọn các phần mềm ghi đọc hệ điều hành cho Raspberry pi 4 (Rufus, balenaEtcher,……). Dưới đây là sử dụng phần mềm Raspberry Pi Imager v1.8.5 để ghi hệ điều hành cho Raspberry pi.



Hình 1. Giao diện ban đầu của ứng dụng Raspberry Pi Imager

- Dưới đây là 1 số cấu hình cơ bản cho Raspberry, sau đó chọn Save và bấm Next chờ cho chương trình ghi hệ điều hành vào thẻ nhớ



Hình 2. Một số cài đặt cơ bản cho Raspberry OS

3.2. Cài đặt các công cụ phát triển (Python, pip, Flask, SQLite, etc.):

- Để cài đặt các công cụ phát triển như Python, Pip, Flask, SQLite, Pandas, mathplotlib, numpy….. ta sử dụng các câu lệnh sau:
    - sudo apt update
    - sudo apt install python3 python3-pip -y (cài đặt Python 3 và pip nếu chưa có sẵn)
    - pip3 install Flask & pip3 install Flask-SQLAlchemy Flask-Migrate (dùng để cài framework web python)
    - sudo apt install sqlite3 -y (cài đặt Sql lite)
    - pip3 install pandas matplotlib numpy (ngoài ra còn 1 số thư viện để phân tích và tạo biểu đồ)

3.3. Cấu hình mạng và truy cập từ xa vào Raspberry Pi:

- Sử dụng open-source RaspAP để cấu hình Raspberry Pi thành Wireless Router giúp quản lí và cấu hình wifi hostpot, có cung cấp giao diện web dễ dàng cấu hình tạo kết nối Internet với các thiết bị khác:
- Các bước cấu hình:

❖ Trước hết ta cần cập nhật danh sách các gói của hệ thống:
    - sudo apt-get update
    - sudo apt-get full-upgrade

❖ Trong Hệ điều hành RPi, mạng không dây 5 GHz bị vô hiệu hóa cho đến khi mã quốc gia này được thiết lập:
    - sudo raspi-config (chọn vùng lãnh thổ)

❖ Để đảm bảo wifi không bị chặn trên Raspberry Pi:

- sudo rfkill unblock wlan

❖ Do lựa chọn bản Debian cài trên Raspberry, để cài dhcph5 ta sử dụng:
- sudo apt-get install dhcpcd5

❖ Cài đặt git, lighttpd, php8, hostapd, dnsmasq và một số gói bổ sung:
- sudo apt-get install lighttpd git hostapd dnsmasq iptables-persistent vnstat qrencode php8.2-cgi jq isoquery

❖ Bật PHP lighttpdvà khởi động lại dịch vụ:
- sudo lighttpd-enable-mod fastcgi-php
- sudo service lighttpd force-reload
- sudo systemctl restart lighttpd.service

❖ Tạo ứng dụng web:
- sudo rm -rf /var/www/html
- sudo git clone https://github.com/RaspAP/raspap-webgui /var/www/html
  WEBROOT="/var/www/html"
  CONFSRC="$WEBROOT/config/50-raspap-router.conf"
  LTROOT=$(grep "server.document-root" /etc/lighttpd/lighttpd.conf | awk -F '=' '{print $2}' | tr -d " \"")

  HTROOT=${WEBROOT/$LTROOT}
  HTROOT=$(echo "$HTROOT" | sed -e 's/\/$//')
  awk "{gsub(\"/REPLACE_ME\",\"$HTROOT\")}1" $CONFSRC > /tmp/50-raspap-router.conf
- sudo cp /tmp/50-raspap-router.conf /etc/lighttpd/conf-available/

❖ Liên kết conf-enabledvà khởi động lại dịch vụ web:
- sudo ln -s /etc/lighttpd/conf-available/50-raspap-router.conf /etc/lighttpd/conf-enabled/50-raspap-router.conf
- sudo systemctl restart lighttpd.service

❖ Sao chép các quy tắc sudoers vào đích của chúng:
- cd /var/www/html
- sudo cp installers/raspap.sudoers /etc/sudoers.d/090_raspap

❖ RaspAP sử dụng một số thư mục để quản lý cấu hình của riêng nó:
- sudo mkdir /etc/raspap/
- sudo mkdir /etc/raspap/backups
- sudo mkdir /etc/raspap/networking
- sudo mkdir /etc/raspap/hostapd
- sudo mkdir /etc/raspap/lighttpd
- sudo mkdir /etc/raspap/system

❖ Thiết lập quyền sở hữu tệp cho www-datangười dùng đối với tệp web và cấu hình RaspAP
- sudo chown -R www-data:www-data /var/www/html
- sudo chown -R www-data:www-data /etc/raspap

❖ RaspAP sử dụng một số tập lệnh shell để quản lý nhiều khía cạnh khác nhau của ứng dụng, bao gồm hostapdghi nhật ký và RaspAP dịch vụ điều khiển RaspAP
- sudo mv installers/enablelog.sh /etc/raspap/hostapd
- sudo mv installers/disablelog.sh /etc/raspap/hostapd
- sudo mv installers/servicestart.sh /etc/raspap/hostapd

- sudo mv installers/debuglog.sh /etc/raspap/system

❖ Thiết lập quyền sở hữu và quyền cho các tập lệnh kiểm soát dịch vụ và ghi nhật ký:
- sudo chown -c root:root /etc/raspap/hostapd/*.sh
- sudo chmod 750 /etc/raspap/hostapd/*.sh
- sudo chown -c root:root /etc/raspap/system/*.sh
- sudo chmod 750 /etc/raspap/system/*.sh

❖ Sao chép và thiết lập quyền sở hữu lighttpd các tập lệnh điều khiển:
- sudo cp installers/configport.sh /etc/raspap/lighttpd
- sudo chown -c root:root /etc/raspap/lighttpd/*.sh

❖ Di chuyển raspapd tệp dịch vụ đến đúng vị trí và kích hoạt nó:
- sudo mv installers/raspapd.service /lib/systemd/system
- sudo systemctl daemon-reload
- sudo systemctl enable raspapd.service

❖ Sao chép các tệp cấu hình cho dhcpcd, dnsmasq, hostapdvà defaults.json. Tùy chọn, sao lưu hostapd.conf
- sudo mv /etc/default/hostapd ~/default_hostapd.old
- sudo cp /etc/hostapd/hostapd.conf ~/hostapd.conf.old
- sudo cp config/hostapd.conf /etc/hostapd/hostapd.conf
- sudo cp config/090_raspap.conf /etc/dnsmasq.d/090_raspap.conf
- sudo cp config/090_wlan0.conf /etc/dnsmasq.d/090_wlan0.conf
- sudo cp config/dhcpcd.conf /etc/dhcpcd.conf
- sudo cp config/config.php /var/www/html/includes/
- sudo cp config/defaults.json /etc/raspap/networking/

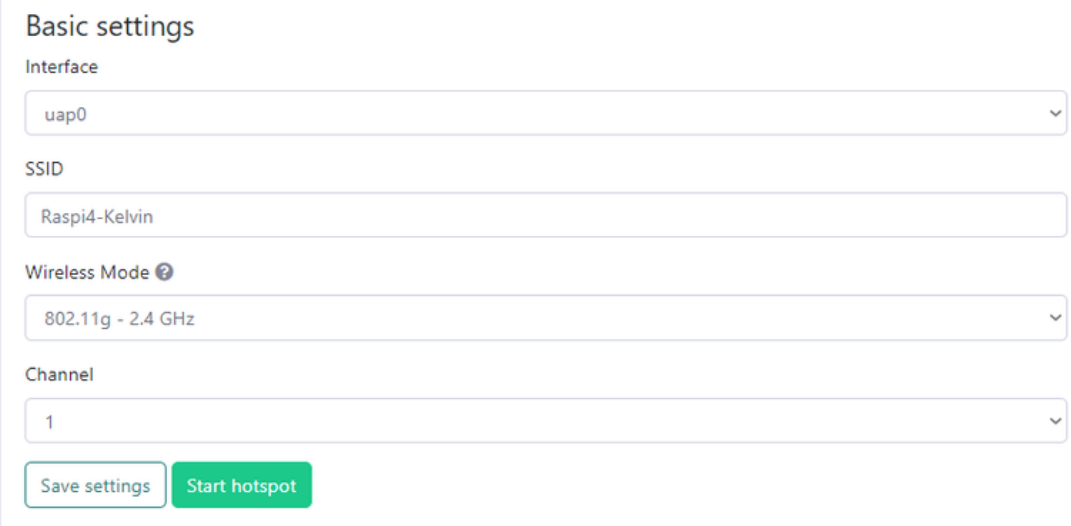
❖ Vô hiệu hóa systemd-networkdvà sao chép cấu hình cầu nối:
- sudo systemctl stop systemd-networkd
- sudo systemctl disable systemd-networkd
- sudo cp config/raspap-bridge-br0.netdev /etc/systemd/network/raspap-bridge-br0.netdev
- sudo cp config/raspap-br0-member-eth0.network /etc/systemd/network/raspap-br0-member-eth0.network

❖ Tối ưu hóa PHP bằng cách sau, thay thế php8.2-cgibằng phiên bản bạn đã cài đặt:
- sudo sed -i -E 's/^session\.cookie_httponly\s*=\s*(0|([O|o]ff)|([F|f]alse)|([N|n]o))\s*$/session.cookie_httponly = 1/' /etc/php/8.2/cgi/php.ini
- sudo sed -i -E 's/^;?opcache\.enable\s*=\s*(0|([O|o]ff)|([F|f]alse)|([N|n]o))\s*$/opcache.enable = 1/' /etc/php/8.2/cgi/php.ini
- sudo phpenmod opcache

❖ Cho phép máy khách WLAN truy cập máy tính trên eth0mạng có dây chính và từ đó truy cập internet.
- echo "net.ipv4.ip_forward=1" | sudo tee /etc/sysctl.d/90_raspap.conf > /dev/null
- sudo sysctl -p /etc/sysctl.d/90_raspap.conf
- sudo /etc/init.d/procps restart

- Cho phép lưu lượng giữa các máy khách trên WLAN và internet, chúng tôi thêm hai iptables quy tắc tường lửa
  - sudo iptables -t nat -A POSTROUTING -j MASQUERADE
  - sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/24 ! -d 192.168.0.0/24 -j MASQUERADE
  - sudo iptables-save | sudo tee /etc/iptables/rules.v4
- Dịch vụ này hostapdbị vô hiệu hóa theo mặc định vì không có cấu hình nào cho nó sau khi cài đặt ban đầu:
  - sudo systemctl unmask hostapd.service
  - sudo systemctl enable hostapd.service
- Cài đặt OpenVPN:
  - sudo apt-get install openvpn
  - sudo sed -i "s/\('RASPI_OPENVPN_ENABLED', \)false/\1true/g" /var/www/html/includes/config.php
  - sudo systemctl enable openvpn-client@client
- Thiết lập quyền sở hữu và quyền hạn:
  - sudo mkdir /etc/raspap/openvpn/
  - sudo cp installers/configauth.sh /etc/raspap/openvpn/
  - sudo chown -c root:root /etc/raspap/openvpn/*.sh
  - sudo chmod 750 /etc/raspap/openvpn/*.sh
- Thêm WireGuard:
  - sudo apt-get install wireguard
  - sudo sed -i "s/\('RASPI_WIREGUARD_ENABLED', \)false/\1true/g" /var/www/html/includes/config.php
  - sudo systemctl enable wg-quick@wg
- Bật tính năng chặn quảng cáo:
  - sudo mkdir /etc/raspap/adblock
  - wget https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts -O /tmp/hostnames.txt
  - wget https://big.oisd.nl/dnsmasq -O /tmp/domains.txt
  - sudo cp /tmp/hostnames.txt /etc/raspap/adblock
  - sudo cp /tmp/domains.txt /etc/raspap/adblock
  - sudo cp installers/update_blocklist.sh /etc/raspap/adblock/
  - sudo chown -c root:www-data /etc/raspap/adblock/*.*
  - sudo chmod 750 /etc/raspap/adblock/*.sh
  - sudo touch /etc/dnsmasq.d/090_adblock.conf
  - echo "conf-file=/etc/raspap/adblock/domains.txt" | sudo tee -a /etc/dnsmasq.d/090_adblock.conf > /dev/null
  - echo "addn-hosts=/etc/raspap/adblock/hostnames.txt" | sudo tee -a /etc/dnsmasq.d/090_adblock.conf > /dev/null
  - sudo sed -i '/dhcp-option=6/d' /etc/dnsmasq.d/090_raspap.conf
  - sudo sed -i "s/\('RASPI_ADBLOCK_ENABLED', \)false/\1true/g" includes/config.php
- Sau cùng là khởi động lại:
  - sudo systemctl reboot

3.4. Cấu hình access point cho Raspberry:



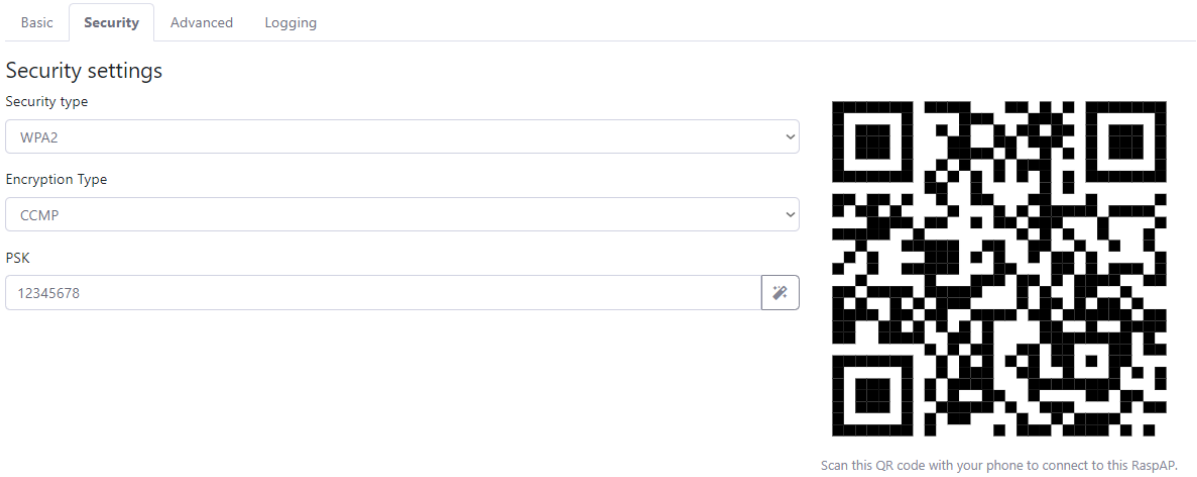
Hình 3. Cấu hình hotspot cơ bản

Các tùy chọn trong phần "Basic settings" cho phép người dùng điều chỉnh những thông số cơ bản của hotspot.

1. **Interface** (uap0):
    a. Interface uap0 là interface ảo giúp tách biệt chức năng AP khỏi interface vật lý (thường là wlan0 có chức năng kết nối tới internet/router), cho phép quản lý AP độc lập. Điều này còn giúp start/stop AP mà không ảnh hưởng đến kết nối internet, dễ dàng debug và giám sát lưu lượng mạng.
    b. Trong trường hợp này, người dùng đang chọn uap0 làm giao diện mạng để phát Wi-Fi.
2. **SSID** (Raspi4-Kelvin):
    a. SSID (Service Set Identifier) là tên của mạng Wi-Fi mà người dùng sẽ nhìn thấy khi quét các mạng Wi-Fi gần đó.
    b. Ở đây, SSID được đặt là "Raspi4-Kelvin", có thể đại diện cho tên của thiết bị Raspberry Pi đang làm điểm phát.
3. **Wireless Mode** (802.11g - 2.4 GHz, có thể phát lên tới băng tần 5Ghz):
    a. Đây là chế độ không dây của mạng Wi-Fi. 802.11g là một trong các chuẩn Wi-Fi, hoạt động trên băng tần 2.4 GHz.
    b. Chuẩn 802.11g cung cấp tốc độ tối đa lên đến 54 Mbps và có khả năng tương thích ngược với chuẩn 802.11b.
    c. Chọn chế độ này có thể giúp tối ưu hóa khả năng tương thích cho các thiết bị cũ hoặc các thiết bị chỉ hỗ trợ 2.4 GHz.
4. **Channel** (1):
    a. Kênh (Channel) là tần số mà mạng Wi-Fi sẽ sử dụng trong băng tần đã chọn. Trong băng tần 2.4 GHz, có 13 kênh khả dụng, với các kênh phổ biến là 1, 6 và 11 để giảm thiểu nhiễu.
    b. Chọn kênh 1 có thể giúp tránh nhiễu nếu các kênh khác (như 6 hoặc 11) đã bị nhiều mạng khác sử dụng.
5. **Save settings**:

a. Nút này dùng để lưu lại các thiết lập mà người dùng đã thay đổi. Sau khi nhấn "Save settings," các cài đặt mới sẽ được áp dụng, nhưng hotspot có thể chưa được kích hoạt ngay.

6. **Start hotspot**:
   a. Nút này dùng để bắt đầu hoặc kích hoạt điểm phát Wi-Fi với các thiết lập đã lưu. Khi nhấn "Start hotspot," thiết bị sẽ bắt đầu phát Wi-Fi với tên mạng, giao diện, chế độ không dây, và kênh đã cấu hình.



Hình 4. Cấu hình mật khẩu cho hotspot

Đây là phần cài đặt bảo mật cho điểm phát Wi-Fi (hotspot). Các tùy chọn trong phần "Security settings" cho phép người dùng thiết lập mức độ bảo mật cho mạng Wi-Fi. Dưới đây là phân tích chi tiết:

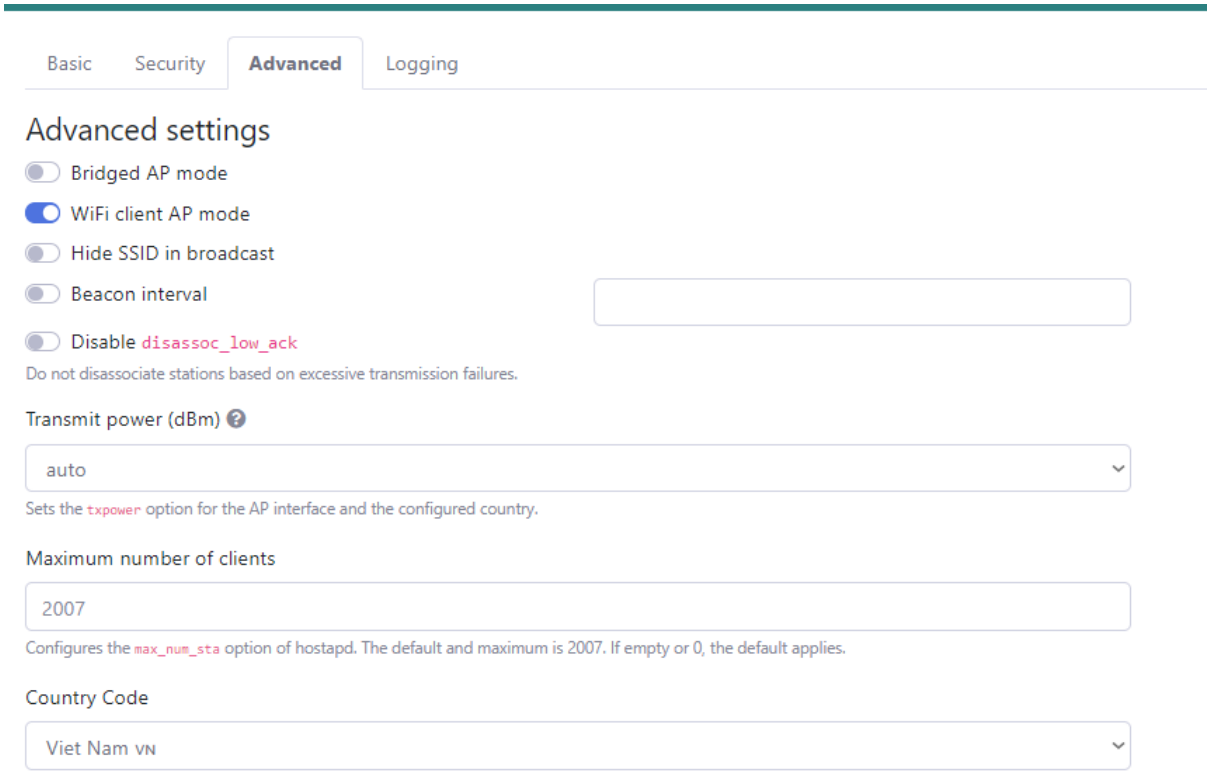
1. **Security type** (WPA2):
   a. Đây là loại bảo mật cho mạng Wi-Fi. WPA2 (Wi-Fi Protected Access 2) là chuẩn bảo mật phổ biến nhất hiện nay, cung cấp mức độ bảo mật cao hơn so với chuẩn WEP hoặc WPA.
   b. WPA2 sử dụng phương pháp mã hóa mạnh mẽ và là lựa chọn tốt để bảo vệ mạng khỏi truy cập trái phép.
2. **Encryption Type** (CCMP):
   a. CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) là loại mã hóa được sử dụng trong WPA2.
   b. CCMP dựa trên chuẩn mã hóa AES (Advanced Encryption Standard), cung cấp khả năng bảo mật cao hơn so với các loại mã hóa cũ như TKIP (Temporal Key Integrity Protocol).
   c. Việc sử dụng CCMP đảm bảo rằng dữ liệu được truyền qua Wi-Fi được mã hóa an toàn, giúp chống lại các tấn công như giả mạo và nghe trộm.
3. **PSK** (12345678):
   a. PSK (Pre-Shared Key) là mật khẩu cho mạng Wi-Fi, cần được nhập khi người dùng kết nối vào mạng.

b. Trong hình, mật khẩu được đặt là 12345678. Đây là mã bảo vệ mạng, nên cần đặt một mật khẩu mạnh hơn để tăng cường bảo mật, đặc biệt là tránh các mật khẩu dễ đoán như ngày tháng hoặc chuỗi số đơn giản.

❖ Ý nghĩa tổng quát

Các thiết lập trong phần "Security settings" này giúp đảm bảo rằng chỉ có những người có mật khẩu đúng mới có thể truy cập vào mạng Wi-Fi. Chọn WPA2 với mã hóa CCMP giúp bảo vệ kết nối một cách hiệu quả. Người dùng nên đặt mật khẩu mạnh và phức tạp hơn để tránh nguy cơ truy cập trái phép.

Ngoài ra, hình ảnh có thể hiển thị một mã QR ở góc phải, có thể giúp người dùng quét mã để kết nối nhanh vào mạng Wi-Fi mà không cần nhập mật khẩu thủ công.



Hình 5. Cấu hình nâng cao cho hotspot

Đây là phần "Advanced settings" trong cài đặt cho điểm phát Wi-Fi, cho phép cấu hình một số thông số nâng cao. Dưới đây là phân tích chi tiết từng tùy chọn:

1. **Bridged AP mode**:
   a. Tùy chọn này cho phép cấu hình chế độ Access Point (AP) dưới dạng "Bridged". Khi được kích hoạt, AP sẽ hoạt động như một cầu nối giữa mạng Wi-Fi và mạng cục bộ, giúp thiết bị kết nối vào mạng Wi-Fi có thể giao tiếp với các thiết bị trong mạng cục bộ.
   b. Cách thực hiện bằng câu lệnh
      - Tạo một cầu nối:

```
sudo brctl addbr br0
sudo brctl addif br0 eth0
```

- Kết nối giao diện AP (uap0) vào cầu nối:
  ```
  sudo brctl addif br0 uap0
  ```

- Sửa file /etc/network/interfaces để duy trì cầu nối sau khi khởi động lại:
  ```
  auto br0
  iface br0 inet dhcp
  bridge_ports eth0 uap0
  ```

2. **WiFi client AP mode**:
   a. Đây là chế độ "WiFi client AP", cho phép thiết bị hoạt động như một trạm (client) và AP cùng lúc. Điều này có nghĩa là thiết bị có thể vừa kết nối vào mạng Wi-Fi khác vừa phát Wi-Fi của chính nó.
   b. Tùy chọn này được bật, nên thiết bị có thể hoạt động đồng thời ở hai chế độ.
   c. Cách thực hiện bằng câu lệnh
      - Kết nối giao diện Wi-Fi đến một mạng khác:

      ```
      sudo nmcli dev wifi connect "SSID_tên_wifi" password "mật_khẩu"
      ```

      - Sửa file cấu hình hostapd.conf để bật AP song song:

      ```
      interface=uap0
      bridge=br0
      ```

3. **Hide SSID in broadcast**:
   a. Tùy chọn này cho phép ẩn tên mạng Wi-Fi (SSID) trong quá trình phát sóng, khiến mạng khó bị phát hiện khi quét Wi-Fi.
   b. Nếu được bật, chỉ những người biết tên mạng mới có thể kết nối vào. Tùy chọn này hiện đang tắt.
   c. Cách thực hiện bằng câu lệnh
      - Thêm hoặc chỉnh sửa trong file cấu hình hostapd.conf:

      ```
      ignore_broadcast_ssid=1
      ```

      - Khởi động lại hostapd:

      ```
      sudo systemctl restart hostapd
      ```

4. **Beacon interval**:
   a. Đây là khoảng thời gian giữa các gói tín hiệu (beacons) mà AP gửi ra để cho các thiết bị biết về sự tồn tại của nó. Giá trị thấp sẽ giúp các thiết bị tìm thấy mạng nhanh hơn nhưng có thể tiêu tốn năng lượng hơn.
   b. Trường này hiện để trống, có nghĩa là hệ thống có thể đang sử dụng giá trị mặc định.
   c. Cách thực hiện bằng câu lệnh
      - Thêm hoặc chỉnh sửa trong file hostapd.conf:

      ```
      beacon_int=100
      ```

Giá trị mặc định là 100ms. Có thể điều chỉnh theo nhu cầu.

5. **Disable disassoc_low_ack**:
    a. Khi bật tùy chọn này, AP sẽ không ngắt kết nối các thiết bị khi phát hiện lỗi truyền tải. Điều này giúp các thiết bị có tín hiệu yếu không bị ngắt kết nối do mất gói dữ liệu thường xuyên.
    b. Tùy chọn này hiện đang tắt, tức là thiết bị có thể ngắt kết nối khi có nhiều lỗi truyền tải.
    c. Cách thực hiện bằng câu lệnh
        - Thêm dòng sau vào file hostapd.conf:

        disassoc_low_ack=0
6. **Transmit power (dBm)** (auto):
    a. Tùy chọn này đặt công suất truyền của AP. Công suất cao hơn sẽ giúp mạng phủ sóng xa hơn, nhưng có thể làm giảm độ ổn định khi có nhiều thiết bị.
    b. Chế độ auto cho phép thiết bị tự động điều chỉnh công suất dựa trên điều kiện hoạt động.
    c. Cách thực hiện bằng câu lệnh
        - Chạy lệnh kiểm tra công suất hiện tại:

        iwconfig uap0
        - Điều chỉnh công suất:

        sudo iwconfig uap0 txpower 20 (Có thể là 20, 40 ,60,.... tuy nhiên giá trị càng cao thì sẽ càng tốn tài nguyên và giảm hiệu suất của Raspberry)
7. **Maximum number of clients** (2007):
    a. Đây là số lượng thiết bị tối đa có thể kết nối vào AP cùng lúc. Giá trị tối đa là 2007, và trong hình cũng đang thiết lập là 2007.
    b. Giá trị này rất cao và không thường được sử dụng trong môi trường gia đình, nhưng có thể thích hợp trong môi trường công cộng hoặc văn phòng lớn.
    c. Cách thực hiện bằng câu lệnh
        - Thêm hoặc chỉnh sửa trong file hostapd.conf:

        max_num_sta=50 (Thay 50 bằng số lượng thiết bị mong muốn.)
8. **Country Code** (Viet Nam vn):
    a. Mã quốc gia giúp xác định các giới hạn tần số và công suất phát sóng hợp pháp cho AP theo quy định của quốc gia.
    b. Thiết lập là "Viet Nam vn", cho phép thiết bị hoạt động tuân theo các quy định tại Việt Nam.
    c. Cách thực hiện bằng câu lệnh
        - Thêm hoặc chỉnh sửa trong file hostapd.conf:

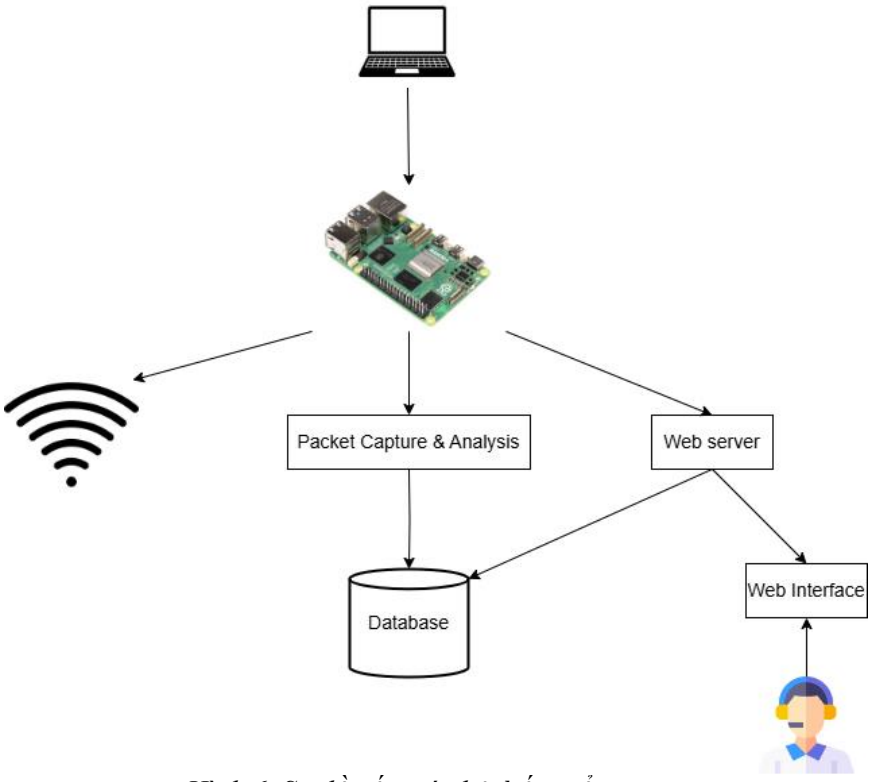
        country_code=VN
        ieee80211d=1
        - Áp dụng mã quốc gia cho giao diện Wi-Fi:

        sudo iw reg set VN

❖ **Ý nghĩa tổng quát**

Phần "Advanced settings" này cung cấp khả năng tùy chỉnh chi tiết cho điểm phát Wi-Fi, phù hợp cho người dùng muốn tối ưu hóa mạng Wi-Fi của mình cho các mục đích cụ thể, như mở rộng phạm vi phủ sóng, giới hạn thiết bị kết nối, hoặc hoạt động trong môi trường đặc thù (như công cộng hoặc văn phòng lớn).

3.5. Xác định kiến trúc tổng quan của ứng dụng:



Hình 6. Sơ đồ cấu trúc hệ thống tổng quan

IV. Phát triển module giám sát lưu lượng mạng

4.1. Giới thiệu về thư viện Scapy và công cụ giám sát lưu lượng mạng:

- Scapy là một thư viện Python mạnh mẽ dành cho việc xử lý và phân tích gói tin mạng. Thư viện này cho phép bạn gửi, chặn, và phân tích các gói tin ở các tầng khác nhau trong mô hình OSI (như Ethernet, IP, TCP, UDP). Chức năng chính của Scapy gồm:
  - Tạo và gửi gói tin: Tạo gói tin tùy chỉnh và gửi chúng qua mạng.
  - Chặn và phân tích gói tin: Bắt và phân tích gói tin, giúp phát hiện các vấn đề bảo mật hoặc hoạt động bất thường.
  - Thực hiện các bài kiểm tra an ninh mạng: Hỗ trợ các kỹ thuật như quét cổng, traceroute, và sniffing để kiểm tra bảo mật mạng.
- Để cài đặt thư viện Scapy, gói bổ sung của thư viện ta sử dụng: **pip3 install scapy**

4.2. Xây dựng chức năng bắt và phân tích gói tin

```python
def packet_callback(packet):
    """Callback function để xử lý gói tin"""
    if IP in packet:
        try:
            # Trích xuất thông tin từ gói tin
            src_ip = packet[IP].src
            dst_ip = packet[IP].dst
            protocol = packet[IP].proto
            length = len(packet)

            # Xác định protocol
            if protocol == 6:  # TCP
                protocol = 'TCP'
                src_port = packet.sport if hasattr(packet, 'sport') else None
                dst_port = packet.dport if hasattr(packet, 'dport') else None
            elif protocol == 17:  # UDP
                protocol = 'UDP'
                src_port = packet.sport if hasattr(packet, 'sport') else None
                dst_port = packet.dport if hasattr(packet, 'dport') else None
            else:
                protocol = 'OTHER'
                src_port = None
                dst_port = None

            # Lưu vào database
            insert_traffic_data(src_ip, dst_ip, protocol, length, src_port, dst_port)

        except Exception as e:
            print(f"Error processing packet: {str(e)}")
```

Hình 7. Hàm phân tích gói tin – network_monitor.py

❖ Hàm packet_callback(packet)

- Chức năng: Hàm callback để xử lý mỗi gói tin được bắt.
- Cách thức:
  - Kiểm tra xem gói tin có chứa lớp IP không (để xác định gói tin có phải là gói tin IP không).
  - Trích xuất các thông tin cơ bản từ gói tin:
    - src_ip: Địa chỉ IP nguồn.
    - dst_ip: Địa chỉ IP đích.
    - protocol: Giao thức (TCP, UDP hoặc khác).
    - length: Kích thước của gói tin.
  - Nếu giao thức là TCP hoặc UDP, nó sẽ lấy thêm thông tin về cổng nguồn (src_port) và cổng đích (dst_port).
  - Nếu là giao thức khác, src_port và dst_port sẽ được đặt là None.
  - Thông tin này sau đó được gửi đến hàm insert_traffic_data() để lưu vào cơ sở dữ liệu.

```python
def start_packet_capture():
    """Bắt đầu bắt gói tin trong thread riêng"""
    try:
        sniff(prn=packet_callback, store=0)
    except Exception as e:
        print(f"Error in packet capture: {str(e)}")
```

Hình 8 .Hàm khởi động giám sát mạng - main.py

❖ Hàm start_packet_capture():Bắt đầu giám sát mạng.

- Cách thức:
  - Hàm sniff() từ thư viện scapy được gọi để bắt gói tin.
  - Hàm sniff() sẽ liên tục bắt các gói tin từ mạng và gọi hàm packet_callback từ network_monitor.py để xử lý mỗi gói tin.
  - Hàm sẽ dừng khi người dùng nhấn Ctrl+C (hoặc có lỗi).

V. Phát triển module phát hiện hoạt động bất thường, phát hiện DoS

5.1 Tạo các điều kiện, ngưỡng phát hiện bất thường của DoS

```python
    def __init__(self):
        # Các ngưỡng phát hiện DDoS
        self.PACKET_THRESHOLD = 1000  # Số lượng gói tin/giây
        self.CONNECTION_THRESHOLD = 100  # Số lượng kết nối đồng thời
        self.TIME_WINDOW = 60  # Cửa sổ thời gian (giây)
        self.BURST_THRESHOLD = 5000  # Ngưỡng burst traffic

        # Lưu trữ thống kê
        self.ip_stats = defaultdict(lambda: {
            'packet_count': 0,
            'last_seen': datetime.now(),
            'connections': set(),
            'bytes_total': 0
        })

        # Danh sách IP đang bị chặn
        self.blocked_ips = set()

        # Lock để thread-safe
        self.stats_lock = threading.Lock()
```

5.2 Lấy dữ liệu traffic:

```python
def get_recent_traffic(self):
    """Lấy dữ liệu traffic gần đây từ database"""
    try:
        conn = sqlite3.connect('network_data.db')
        query = """
        SELECT timestamp, src_ip, dst_ip, protocol, length, src_port, dst_port
        FROM network_traffic
        WHERE timestamp >= datetime('now', '-1 minute')
        """
        df = pd.read_sql_query(query, conn)
        conn.close()
        return df
    except Exception as e:
        logging.error(f"Error getting traffic data: {e}")
        return pd.DataFrame()
```

Hình 9. Hàm cập nhật dữ liệu lưu lượng mạng mới nhất từ database - ddos_detection.py

- **Chức năng**:
  - Kết nối tới cơ sở dữ liệu SQLite (network_data.db) và lấy dữ liệu traffic từ bảng network_traffic trong vòng 1 phút gần đây.
  - Trả về dữ liệu dưới dạng DataFrame của Pandas để dễ dàng phân tích.
- **Cách thức**:
  - Chạy lệnh SQL để lấy các cột: thời gian (timestamp), địa chỉ IP nguồn và đích, giao thức, kích thước gói tin (length), và các cổng.

5.3 Phân tích mẫu traffic:

```python
def analyze_traffic_patterns(self, df):
    """Phân tích mẫu traffic để phát hiện DoS"""
    if df.empty:
        return

    current_time = datetime.now()
    alerts = []

    with self.stats_lock:
        # Phân tích theo source IP
        for src_ip in df['src_ip'].unique():
            ip_data = df[df['src_ip'] == src_ip]

            # Tính toán các metric
            packets_per_second = len(ip_data) / self.TIME_WINDOW
            unique_connections = len(set(zip(ip_data['dst_ip'], ip_data['dst_port'])))
            total_bytes = ip_data['length'].sum()

            # Cập nhật thống kê
            self.ip_stats[src_ip]['packet_count'] = packets_per_second
            self.ip_stats[src_ip]['connections'].update(
                set(zip(ip_data['dst_ip'], ip_data['dst_port']))
            )
            self.ip_stats[src_ip]['bytes_total'] = total_bytes
            self.ip_stats[src_ip]['last_seen'] = current_time

            # Kiểm tra các dấu hiệu DoS
            if self._check_ddos_indicators(src_ip):
                alert_msg = self._generate_alert(src_ip)
                alerts.append(alert_msg)
                self.blocked_ips.add(src_ip)

    return alerts
```

Hình 10. Hàm phân tích mẫu lưu lượng để phát hiện tấn công DoS - ddos_detection.py

- **Chức năng**:

Xử lý dữ liệu từ DataFrame để kiểm tra xem có dấu hiệu tấn công DoS hay không.

- **Cách thức:**
  - **Lặp qua từng IP nguồn (src_ip)**:
  - Tính các chỉ số:
    - packets_per_second: Tần suất gói tin (gói/giây).
    - unique_connections: Số kết nối duy nhất (IP đích + cổng đích).
    - total_bytes: Tổng lượng dữ liệu (bytes) được gửi.
  - Cập nhật thông tin vào self.ip_stats:
    - Lưu số lượng gói, số lượng kết nối, tổng byte và thời gian cuối cùng.
  - **Kiểm tra điều kiện DoS**:
    - Nếu IP vượt qua ngưỡng (packet rate, số kết nối, hoặc burst traffic), gọi _check_ddos_indicators để xác minh.

- Nếu phát hiện DoS, tạo cảnh báo và thêm IP vào danh sách blocked_ips.

5.4 Kiểm tra dấu hiệu DoS:

```python
def _check_ddos_indicators(self, ip):
    """Kiểm tra các dấu hiệu DDoS cho một IP"""
    stats = self.ip_stats[ip]

    # Kiểm tra các điều kiện
    high_packet_rate = stats['packet_count'] > self.PACKET_THRESHOLD
    many_connections = len(stats['connections']) > self.CONNECTION_THRESHOLD
    high_bandwidth = stats['bytes_total'] > self.BURST_THRESHOLD

    # Trả về True nếu thỏa mãn ít nhất 2 điều kiện
    return sum([high_packet_rate, many_connections, high_bandwidth]) >= 2
```

Hình 11. Hàm kiểm tra các dấu hiệu DoS bằng các điều kiện ở 5.1 - ddos_detection.py

Chức năng:

- Xác định xem một IP có phải là nguồn tấn công DoS hay không bằng cách kiểm tra 3 điều kiện:
  - Tần suất gói tin cao hơn ngưỡng (high_packet_rate).
  - Số kết nối đồng thời vượt ngưỡng (many_connections).
  - Băng thông cao hơn ngưỡng (high_bandwidth).
- Nếu ít nhất 2 trong 3 điều kiện thỏa mãn, trả về True.

5.4 Tạo cảnh báo:

```python
    def _generate_alert(self, ip):
        """Tạo thông báo cảnh báo chi tiết"""
        stats = self.ip_stats[ip]
        alert = f"""
DOS ATTACK DETECTED!
Source IP: {ip}
Time: {datetime.now()}
Indicators:
- Packets/sec: {stats['packet_count']:.2f}
- Unique connections: {len(stats['connections'])}
- Total bandwidth: {stats['bytes_total']/1024:.2f} KB
"""
        logging.warning(alert)
        return alert

    def cleanup_old_stats(self):
        """Xóa thống kê cũ"""
        current_time = datetime.now()
        with self.stats_lock:
            for ip in list(self.ip_stats.keys()):
                if (current_time - self.ip_stats[ip]['last_seen']).seconds > self.TIME_WINDOW:
                    del self.ip_stats[ip]
```

Hình 12. Tạo các cảnh báo chi tiết về tấn công DoS - ddos_detection.py

- **Chức năng**:
  - Tạo một thông báo chi tiết khi phát hiện tấn công DoS, bao gồm:
    - Địa chỉ IP nguồn.
    - Tần suất gói tin.
    - Số kết nối duy nhất.
    - Tổng băng thông.
- **Ghi log**: Cảnh báo được lưu vào file static/ddos_alerts.log.

5.5 Dọn dẹp dữ liệu cũ:

```
def cleanup_old_stats(self):
    """Xóa thống kê cũ"""
    current_time = datetime.now()
    with self.stats_lock:
        for ip in list(self.ip_stats.keys()):
            if (current_time - self.ip_stats[ip]['last_seen']).seconds > self.TIME_WINDOW:
                del self.ip_stats[ip]
```

Hình 13. Xóa dữ liệu về lưu lượng cũ - ddos_detection.py

Chức năng:

- Xóa dữ liệu thống kê của các IP không hoạt động trong TIME_WINDOW (60 giây).

5.6 Bắt đầu giám sát:

```python
def start_ddos_monitoring():
    """Khởi động monitoring trong thread riêng"""
    detector = DDoSDetector()

    def monitoring_task():
        while True:
            try:
                # Lấy và phân tích traffic
                df = detector.get_recent_traffic()
                alerts = detector.analyze_traffic_patterns(df)

                # Ghi alerts vào file
                if alerts:
                    with open('static/ddos_alerts.txt', 'w') as f:
                        for alert in alerts:
                            f.write(f"{alert}\n{'=' * 50}\n")

                # Dọn dẹp thống kê cũ
                detector.cleanup_old_stats()

                time.sleep(5)

            except Exception as e:
                logging.error(f"Error in DDoS monitoring: {e}")
                time.sleep(5)

    # Khởi động thread monitoring
    monitor_thread = threading.Thread(target=monitoring_task, daemon=True)
    monitor_thread.start()
    return detector
```

Hình 14: Hàm bắt đầu giám sát DoS

Chức năng:

- Bắt đầu giám sát và phân tích, sau đó ghi vào file ddos_alert.txt

VI. Xây dựng hệ thống lưu trữ và phân tích log

6.1. Sử dụng SQLite để lưu trữ dữ liệu mạng và cảnh báo:

```python
def init_db():
    """Khởi tạo database"""
    try:
        conn = sqlite3.connect('network_data.db')
        cursor = conn.cursor()

        # Tạo bảng network_traffic nếu chưa tồn tại
        cursor.execute('''
        CREATE TABLE IF NOT EXISTS network_traffic (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
            src_ip TEXT,
            dst_ip TEXT,
            protocol TEXT,
            length INTEGER,
            src_port INTEGER,
            dst_port INTEGER
        )
        ''')

        # Tạo index cho các trường thường được truy vấn
        cursor.execute('CREATE INDEX IF NOT EXISTS idx_timestamp ON network_traffic(timestamp)')
        cursor.execute('CREATE INDEX IF NOT EXISTS idx_src_ip ON network_traffic(src_ip)')
        cursor.execute('CREATE INDEX IF NOT EXISTS idx_dst_ip ON network_traffic(dst_ip)')

        conn.commit()
        conn.close()
        print("Database initialized successfully")
    except Exception as e:
        print(f"Error initializing database: {str(e)}")
```

Hình 15. Hàm khởi tạo cở sở dữ liệu nếu chưa tồn tại – database.py

❖ Hàm init_db() Hàm này thiết lập cấu trúc cơ sở dữ liệu và bảng network_traffic nếu chưa tồn tại.

- Tạo bảng network_traffic: Sử dụng câu lệnh CREATE TABLE IF NOT EXISTS để tạo bảng network_traffic với các cột:
- id: Khóa chính tự động tăng.
- timestamp: Thời gian tạo bản ghi (dạng DATETIME, mặc định là thời gian hiện tại).
- src_ip: Địa chỉ IP nguồn.
- dst_ip: Địa chỉ IP đích.
- protocol: Giao thức (dạng TEXT để lưu tên hoặc mã giao thức).
- length: Kích thước gói tin.
- src_port: Cổng nguồn.
- dst_port: Cổng đích.
- Tạo chỉ mục: Đoạn mã tạo các chỉ mục cho các trường timestamp, src_ip, và dst_ip, giúp tăng tốc quá trình truy vấn khi tìm kiếm các bản ghi theo các trường này.

```python
def insert_traffic_data(src_ip, dst_ip, protocol, length, src_port=None, dst_port=None):
    """Thêm dữ liệu traffic vào database"""
    try:
        conn = sqlite3.connect('network_data.db')
        cursor = conn.cursor()

        cursor.execute('''
        INSERT INTO network_traffic (timestamp, src_ip, dst_ip, protocol, length, src_port, dst_port)
        VALUES (?, ?, ?, ?, ?, ?, ?)
        ''', (datetime.now(), src_ip, dst_ip, protocol, length, src_port, dst_port))

        conn.commit()
        conn.close()
    except Exception as e:
        print(f"Error inserting traffic data: {str(e)}")
```

Hình 16. Hàm thêm dữ liệu lưu lượng mạng vào bảng database - database.py

❖ Hàm insert_traffic_data(): Chèn dữ liệu lưu lượng mạng vào bảng network_traffic.

- Cách thức:
  - Tham số: Địa chỉ IP nguồn, địa chỉ IP đích, giao thức, kích thước gói tin và tùy chọn cổng nguồn, cổng đích.
  - Hành động: Mở kết nối cơ sở dữ liệu, chèn dữ liệu lưu lượng với thời gian hiện tại, cam kết thay đổi và đóng kết nối.

```python
def get_recent_traffic(limit=100):
    """Lấy dữ liệu traffic gần đây nhất"""
    try:
        conn = sqlite3.connect('network_data.db')
        cursor = conn.cursor()

        cursor.execute('''
        SELECT timestamp, src_ip, dst_ip, protocol, length
        FROM network_traffic
        ORDER BY timestamp DESC
        LIMIT ?
        ''', (limit,))

        data = cursor.fetchall()
        conn.close()

        return data
    except Exception as e:
        print(f"Error getting recent traffic: {str(e)}")
        return []
```

Hình 17. Hàm cập nhật dữ liệu mới gần nhất - database.py

❖ Hàm get_recent_traffic():Lấy dữ liệu lưu lượng gần đây nhất từ cơ sở dữ liệu.

- Cách thức:
  - Tham số: Giới hạn số lượng bản ghi cần lấy (mặc định là 100).
  - Hành động: Lấy các bản ghi lưu lượng gần đây nhất theo thứ tự thời gian, sau đó đóng kết nối.

```python
def clear_old_data(days=7):
    """Xóa dữ liệu cũ hơn số ngày chỉ định"""
    try:
        conn = sqlite3.connect('network_data.db')
        cursor = conn.cursor()

        cursor.execute('''
        DELETE FROM network_traffic
        WHERE timestamp < datetime('now', '-? days')
        ''', (days,))

        conn.commit()
        conn.close()
        print(f"Cleared data older than {days} days")
    except Exception as e:
        print(f"Error clearing old data: {str(e)}")
```

Hình 18. Xóa dữ liệu cũ sau khoảng thời gian 7 ngày - database.py

❖ Hàm clear_old_data(): Xóa dữ liệu lưu lượng cũ hơn một số ngày nhất định (mặc định là 7 ngày).

- Cách thức:
  - Tham số: Số ngày cần giữ lại dữ liệu (mặc định là 7).
  - Hành động: Xóa các bản ghi trong cơ sở dữ liệu có thời gian cũ hơn số ngày đã chỉ định và cam kết thay đổi.

6.2. Tạo các truy vấn để phân tích xu hướng và tạo báo cáo

6.2.1. Phân tích lưu lượng đáng ngờ và tạo báo cáo

```python
def get_data_from_db():
    """Đọc dữ liệu từ SQLite database"""
    try:
        conn = sqlite3.connect('network_data.db')
        query = """
        SELECT timestamp, src_ip, dst_ip, protocol, length, src_port, dst_port
        FROM network_traffic
        ORDER BY timestamp DESC
        LIMIT 10000
        """
        df = pd.read_sql_query(query, conn)
        conn.close()
        return df
    except Exception as e:
        print(f"Error reading from database: {e}")
        return pd.DataFrame()
```

```python
def analyze_traffic():
    """Phân tích traffic và tạo báo cáo"""
    df = get_data_from_db()

    if df.empty:
        print("No data available for analysis")
        # Tạo file thống kê trống
        with open('static/traffic_stats.txt', 'w') as f:
            f.write("No traffic data available for analysis")
        with open('static/nmap_detections.txt', 'w') as f:
            f.write("No data available for scan detection")
        with open('static/ddos_detections.txt', 'w') as f:
            f.write("No data available for DDoS detection")
        return
```

Hình 19. Hàm phân tích các lưu lượng đáng ngờ - log_analysis.py

❖ Hàm analyze_traffic():

- Sử dụng hàm get_data_from_db() để lấy dữ liệu từ cơ sở dữ liệu (dự đoán là một DataFrame pandas).
- Nếu không có dữ liệu, tạo các tệp thống kê trống (traffic_stats.txt, nmap_detections.txt, ddos_detections.txt) và kết thúc.

```python
# Chuyển đổi timestamp thành datetime
df['timestamp'] = pd.to_datetime(df['timestamp'])

try:
    # Tạo figure mới với kích thước lớn hơn
    plt.figure(figsize=(15, 10))

    # 1. Traffic theo thời gian
    plt.subplot(2, 2, 1)
    traffic_by_hour = df.groupby(df['timestamp'].dt.hour)['length'].mean()
    traffic_by_hour.plot(kind='bar')
    plt.title('Average Traffic by Hour')
    plt.xlabel('Hour')
    plt.ylabel('Average Packet Size')
    plt.xticks(rotation=45)

    # 2. Protocol distribution
    plt.subplot(2, 2, 2)
    protocol_counts = df['protocol'].value_counts()
    plt.pie(protocol_counts.values, labels=protocol_counts.index, autopct='%1.1f%%')
    plt.title('Protocol Distribution')
```

Hình 20. Hàm phân tích biểu đồ theo thời gian

```python
# 2. Protocol distribution
plt.subplot(2, 2, 2)
protocol_counts = df['protocol'].value_counts()
plt.pie(protocol_counts.values, labels=protocol_counts.index, autopct='%1.1f%%')
plt.title('Protocol Distribution')
```

Hình 21. Hàm phân tích biểu đồ theo giao thức

```
    # 3. Top source IPs
    plt.subplot(2, 2, 3)
    top_ips = df['src_ip'].value_counts().head(10)
    top_ips.plot(kind='bar')
    plt.title('Top 10 Source IPs')
    plt.xticks(rotation=45)
    plt.xlabel('Source IP')
    plt.ylabel('Packet Count')
```

Hình 22. Hàm phân tích biểu đồ top các IP

```
    # 4. Packet size distribution
    plt.subplot(2, 2, 4)
    plt.hist(df['length'], bins=50)
    plt.title('Packet Size Distribution')
    plt.xlabel('Packet Size')
    plt.ylabel('Frequency')

    # Điều chỉnh layout để tránh chồng chéo
    plt.tight_layout(pad=3.0)

    # Lưu biểu đồ
    plt.savefig('static/traffic_analysis.png', bbox_inches='tight', dpi=300)
    plt.close()

except Exception as e:
    print(f"Error creating plots: {e}")
    # Tạo biểu đồ lỗi
    plt.figure(figsize=(10, 6))
    plt.text(0.5, 0.5, f'Error creating analysis plots: {str(e)}',
             ha='center', va='center', wrap=True)
    plt.savefig('static/traffic_analysis.png')
    plt.close()

# Tạo báo cáo thống kê
try:
    stats = [
        f"Analysis Time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}",
        f"Total Packets: {len(df)}",
        f"Unique Source IPs: {df['src_ip'].nunique()}",
        f"Unique Destination IPs: {df['dst_ip'].nunique()}",
        f"Average Packet Size: {df['length'].mean():.2f} bytes",
        f"Most Common Protocol: {df['protocol'].mode().iloc[0] if not df.empty else 'N/A'}",
        "\nTop 5 Source IPs:",
        df['src_ip'].value_counts().head().to_string(),
        "\nProtocol Distribution:",
        df['protocol'].value_counts().to_string()
    ]

    with open('static/traffic_stats.txt', 'w') as f:
        f.write('\n'.join(stats))
```

Hình 23. Hàm phân tích kích thước gói vẽ - log_analysis.py

❖ Phân tích và tạo biểu đồ:
- Biểu đồ 1: Lưu lượng theo giờ: Tính toán và vẽ biểu đồ bar thể hiện kích thước gói tin trung bình theo từng giờ.
- Biểu đồ 2: Phân bố giao thức: Vẽ biểu đồ pie để thể hiện phân bố các giao thức trong dữ liệu.
- Biểu đồ 3: Top 10 địa chỉ IP nguồn: Hiển thị biểu đồ bar của 10 địa chỉ IP nguồn có nhiều gói tin nhất.
- Biểu đồ 4: Phân bố kích thước gói tin: Vẽ histogram để thể hiện phân bố kích thước gói tin.

❖ Tạo báo cáo thống kê:
- Tạo báo cáo thống kê với các thông tin như:
  - Thời gian phân tích
  - Tổng số gói tin
  - Số lượng địa chỉ IP nguồn và đích duy nhất
  - Kích thước gói tin trung bình
  - Giao thức phổ biến nhất
  - Top 5 địa chỉ IP nguồn
  - Phân bố giao thức
- Báo cáo được lưu vào tệp traffic_stats.txt.

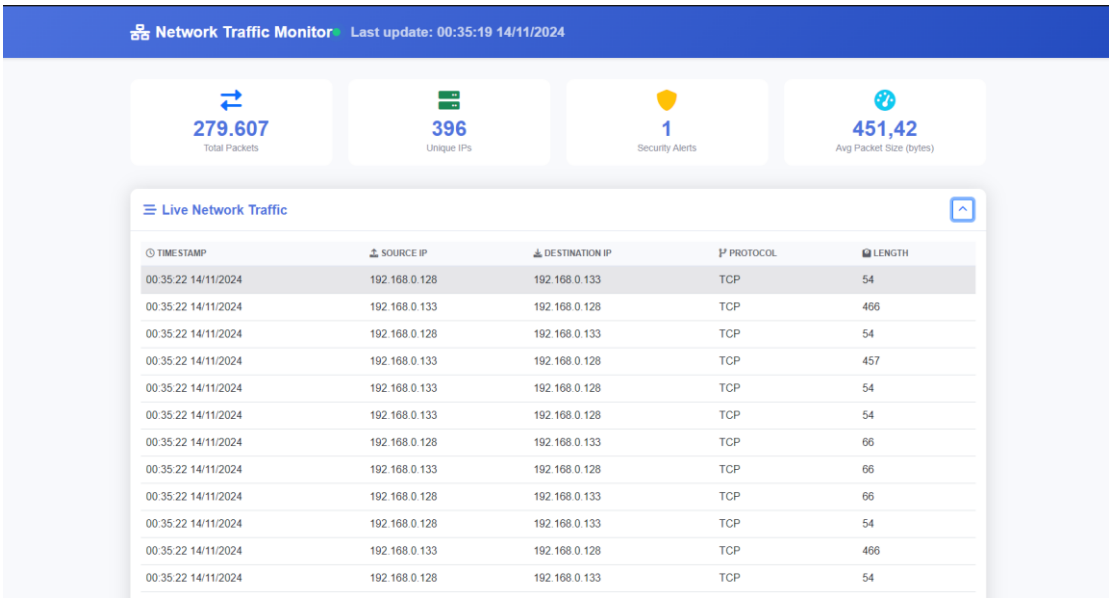
VII. Phát triển giao diện web

7.1. Giới thiệu về Flask và ứng dụng web server trên Raspberry Pi:

- Flask là một microframework web nhẹ và linh hoạt, được viết bằng Python. Nó rất dễ sử dụng, hỗ trợ các tính năng như kết xuất template HTML, tạo API, và dễ dàng mở rộng với các tiện ích mở rộng
- rong dự án này, Flask sẽ được sử dụng để xây dựng một web server trên Raspberry Pi. Các chức năng chính bao gồm:
  - Giao diện web hiển thị thông tin mạng và cảnh báo.
  - Trang cấu hình cho phép người dùng điều chỉnh các tham số giám sát và cảnh báo.
  - Tích hợp với Scapy và SQLite để phân tích lưu lượng mạng và lưu trữ log.

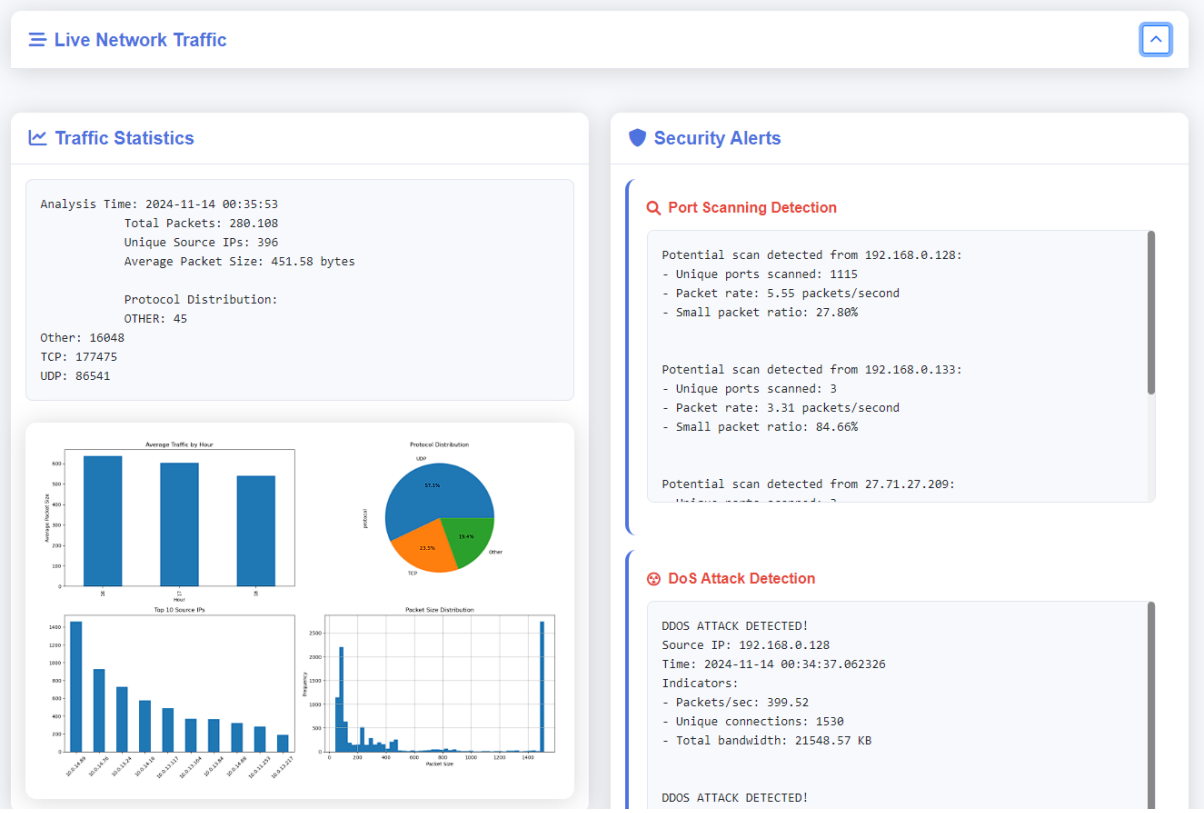
7.2. Thiết kế và phát triển dashboard hiển thị thông tin tổng quan:

7.2.1. Hiển thị thông tin thời gian thực về lưu lượng mạng:

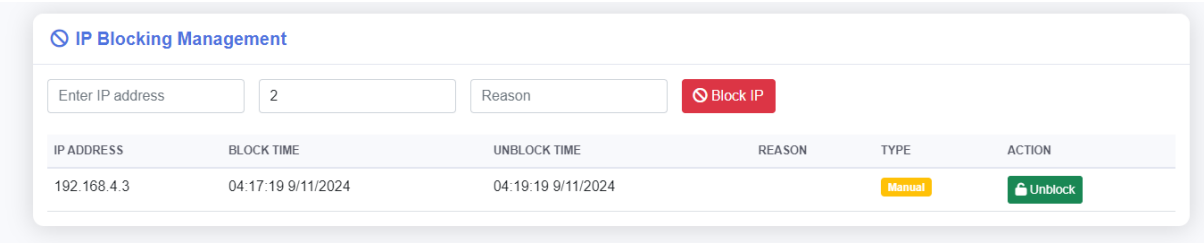


Hình 24. Giao diện theo dõi lưu lượng mạng thời gian thực

7.2.2. Đồ thị và biểu đồ về lưu lượng và cảnh báo:



Hình 25. Giao diện hiển thị thông kê lưu lượng, cảnh báo bảo mật



Hình 26. Giao diện thao tác chức năng Block, Unblock các IP

IX. Kết luận và hướng phát triển

9.1. Tóm tắt kết quả đạt được:

- Hệ thống Access Point (AP) hoạt động ổn định:
  - Raspberry Pi đã được cấu hình thành một điểm phát Wi-Fi hoạt động với dịch vụ DHCP và DNS (dnsmasq) để phân phối địa chỉ IP cho các thiết bị kết nối.
- Giao diện quản lý web GUI của RaspAP hoạt động tốt:

- Giao diện web giúp người dùng dễ dàng cấu hình AP mà không cần dòng lệnh.

9.2. Các vấn đề còn tồn đọng:

- Hiệu năng hạn chế khi lưu lượng truy cập lớn:
  - Raspberry Pi có hạn chế về tài nguyên phần cứng, có thể gây ra hiện tượng chậm hoặc không đáp ứng khi có nhiều thiết bị truy cập cùng lúc.
- Khả năng mở rộng giao diện web còn hạn chế:
  - Giao diện RaspAP hiện tại chủ yếu phục vụ quản lý cơ bản. Các tính năng nâng cao như giám sát thời gian thực, phân tích lưu lượng mạng còn thiếu.
- Hạn chế trong phát hiện lỗi hoặc bất thường:
  - Hiện tại, hệ thống không có tính năng tự động phát hiện và cảnh báo các hoạt động bất thường (như tấn công mạng, lưu lượng lớn bất thường).
- Hạn chế về giao diện người dùng:
  - RaspAP chưa thân thiện với các thiết bị di động hoặc màn hình nhỏ, cần cải thiện giao diện để sử dụng dễ dàng hơn.
- Biểu đồ thể hiện lưu lượng:
  - Biểu đồ thể hiện lưu lượng còn bị lỗi khi cập nhật lại lưu lượng.

9.3. Hướng phát triển trong tương lai:

- Sử dụng mô hình Machine Learning để phát hiện các hoạt động bất thường như:
  - Port scanning (quét cổng).
  - Tấn công DoS hoặc brute-force.
  - Các hành vi truy cập không hợp lệ (sử dụng mẫu lưu lượng bất thường để phát hiện).
- Hệ thống có thể gửi thông báo qua email hoặc giao diện web khi phát hiện sự cố.
- Tối ưu hiệu năng để có thể sử dụng Raspberry Pi một cách mượt mà hơn, tìm ra giải pháp phù hợp để đưa các tính năng liên quan lên Cloud để giảm tải cho Raspberry Pi.
- Phát triển web interface thân thiện với người dùng hơn. Có khả năng triển khai ở doanh nghiệp vừa và nhỏ.

X. Tham khảo:

Tài liệu tham khảo

[1] "Hướng dẫn bật tính năng mạng không dây," Tài liệu RaspAP. https://docs.raspap.com/manual/#enable-wireless-operation.

[2] "Cơ bản về Access Point," Tài liệu RaspAP. Có sẵn tại: https://docs.raspap.com/ap-basics.

[3] Diễn đàn Raspberry Pi, "Vấn đề RaspAP không hoạt động ngay sau khi cài đặt,". https://forums.raspberrypi.com/viewtopic.php?t=322171.

[4] Kho lưu trữ GitHub của RaspAP, "RaspAP WebGUI,". https://github.com/raspap/raspap-webgui.

[5] Raspberry Pi Stack Exchange, "Khắc phục sự cố thiết lập RaspAP,". https://raspberrypi.stackexchange.com/questions/132194/raspap-not-working-out-of-the-box.

[6] Thảo luận trên GitHub của RaspAP, "Phản hồi và giải đáp của cộng đồng,".https://github.com/raspap/raspap-webgui/discussions.