

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



MÔN HỌC: AN TOÀN CHO IOT
Đề tài: Mô phỏng Lightweight IDS trên Raspberry Pi

Giảng viên: TS. Phạm Văn Hưởng
Học viên thực hiện: Phạm Quang Long

Hà Nội, 2024

MỤC LỤC

Danh mục hình ảnh.....	3
Lời nói đầu.....	4
Chương 1. Cơ sở lý thuyết.....	5
1.1. Giới thiệu Raspberry Pi	5
1.2. Giới thiệu Lightweight IDS	6
1.3. Giới thiệu Snort	7
Chương 2. Phân tích, thiết kế hệ thống	9
2.1. Giới thiệu	9
2.2. Sơ đồ thiết kế.....	10
Chương 3. Triển khai hệ thống.....	15
3.1. Triển khai hệ thống.....	15
3.2. Lập trình cảnh báo	18
3.3. Kết quả, đánh giá kết quả	20
Kết luận.....	22
Tài liệu tham khảo	23

DANH MỤC HÌNH ẢNH

Hình 1. Mô hình mạng thực tế thử nghiệm	10
Hình 2. Raspberry Pi 4 Model B	12
Hình 3. Router	13
Hình 4. Switch	14
Hình 5. Mô hình mạng đề xuất sử dụng	14
Hình 6. Cài đặt wireshark	16
Hình 7. Cài đặt tshark	16
Hình 8. Cài đặt nmap	17
Hình 9. Cài đặt snort	17
Hình 10. Cấu hình snort	18
Hình 11. Kết quả	20

LỜI NÓI ĐẦU

Trong thời đại mà các mối đe dọa an ninh máy tính lan tràn, việc bảo vệ hệ thống mạng chống lại các cuộc tấn công nghe lén và rình mò (man-in-the-middle) trở nên quan trọng hơn bao giờ hết. Hệ thống này giải quyết những thách thức mà các cá nhân không am hiểu công nghệ phải đối mặt khi cố gắng hiểu và sử dụng các công cụ bảo mật mạng mã nguồn mở truyền thống. Mục tiêu là tạo ra một bộ công cụ mạng có thể mở rộng, dễ tiếp cận cho tất cả người dùng, cung cấp một giải pháp nhẹ nhưng hiệu quả để phát hiện và giảm thiểu các cuộc tấn công nghe lén. Hệ thống được thiết kế chạy trên Raspberry Pi Phiên bản 3B và các model mới hơn, cũng như bất kỳ máy nào dựa trên Debian với phần cứng tương đương hoặc tốt hơn.

Sau thời gian thực hiện, các mục tiêu đặt ra khi thực hiện đề tài cơ bản đã đạt được và được bố cục thành ba chương, gồm:

- **Chương 1. Cơ sở lý thuyết:** chương này đã mô tả tổng quan về đề tài cũng như nêu lên một số các lý thuyết xoay quanh bài toán.
- **Chương 2. Phân tích, thiết kế hệ thống:** mô tả chi tiết thiết kế hệ thống
- **Chương 3. Triển khai hệ thống:** chương này tập trung mô tả các kết quả đạt được của đề tài, đánh giá kết quả và đưa ra phương hướng phát triển của đề tài.

Do vốn kiến thức và kinh nghiệm còn hạn hẹp nên đề tài không tránh khỏi có những sai sót. Em rất mong nhận được sự góp ý của thầy cô và các bạn.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu Raspberry Pi

Raspberry Pi là một dòng máy tính nhỏ gọn, giá rẻ, có kích thước chỉ bằng một chiếc thẻ tín dụng, được phát triển bởi Raspberry Pi Foundation tại Anh. Mục tiêu ban đầu của dự án là thúc đẩy việc giảng dạy khoa học máy tính trong trường học và các nước đang phát triển, nhưng Raspberry Pi đã nhanh chóng trở thành một công cụ phổ biến cho các dự án DIY, ứng dụng IoT, và nhiều mục đích khác.

Các đặc điểm nổi bật:

- **Giá rẻ:** Với mức giá chỉ từ vài chục đô la, Raspberry Pi mang đến một giải pháp tính toán mạnh mẽ mà không cần đầu tư quá nhiều.
- **Kích thước nhỏ gọn:** Kích thước nhỏ gọn giúp Raspberry Pi dễ dàng tích hợp vào các dự án khác nhau, từ robot đến hệ thống tự động hóa gia đình.
- **Khả năng tùy biến cao:** Raspberry Pi hỗ trợ nhiều hệ điều hành khác nhau, chủ yếu là các bản phân phối Linux, cho phép bạn tùy chỉnh và cài đặt các phần mềm theo nhu cầu của mình.
- **Cộng đồng lớn mạnh:** Raspberry Pi có một cộng đồng người dùng và nhà phát triển đông đảo, sẵn sàng chia sẻ kiến thức, kinh nghiệm và hỗ trợ lẫn nhau.

Các ứng dụng phổ biến:

- **Học tập và giáo dục:** Raspberry Pi là một công cụ tuyệt vời để học lập trình, điện tử, và các kỹ năng STEM khác.
- **Dự án DIY:** Với Raspberry Pi, bạn có thể xây dựng các dự án thú vị như máy chơi game retro, trung tâm media, hệ thống giám sát an ninh, và nhiều hơn nữa.
- **IoT (Internet of Things):** Raspberry Pi có thể được sử dụng để kết nối các thiết bị với nhau và với internet, tạo ra các ứng dụng IoT thông minh.
- **Máy chủ web và ứng dụng:** Raspberry Pi có thể hoạt động như một máy chủ web nhỏ gọn, chạy các ứng dụng web đơn giản hoặc lưu trữ dữ liệu.
- **Máy tính để bàn giá rẻ:** Với các phiên bản Raspberry Pi mạnh mẽ hơn, bạn có thể sử dụng nó như một máy tính để bàn cơ bản cho các tác vụ văn phòng, duyệt web, và giải trí.

Các phiên bản Raspberry Pi:

Raspberry Pi có nhiều phiên bản khác nhau với các mức giá và tính năng khác nhau. Một số phiên bản phổ biến bao gồm:

- **Raspberry Pi Pico:** Một vi điều khiển nhỏ gọn và giá rẻ, phù hợp cho các dự án điện tử đơn giản.
- **Raspberry Pi Zero:** Một phiên bản siêu nhỏ gọn và tiết kiệm năng lượng, lý tưởng cho các dự án nhúng.
- **Raspberry Pi 3 và 4:** Các phiên bản mạnh mẽ hơn, phù hợp cho các ứng dụng đòi hỏi hiệu năng cao hơn như máy chủ web, máy tính để bàn, và học máy.

1.2. Giới thiệu Lightweight IDS

Lightweight Intrusion Detection System (LWIDS) hay còn gọi là Hệ thống Phát hiện Xâm nhập Hạng nhẹ, là một giải pháp bảo mật mạng được thiết kế để phát hiện các hoạt động đáng ngờ hoặc độc hại trên mạng một cách chủ động, đồng thời giảm thiểu tác động đến hiệu suất hệ thống. LWIDS hoạt động bằng cách giám sát và phân tích lưu lượng mạng, so sánh nó với các mẫu tấn công đã biết và các hành vi bất thường, sau đó cung cấp cảnh báo kịp thời cho quản trị viên hệ thống.

Ưu điểm của LWIDS:

- **Phát hiện tấn công chủ động:** LWIDS có khả năng phát hiện các cuộc tấn công mạng một cách chủ động, giúp quản trị viên có thể phản ứng kịp thời để ngăn chặn thiệt hại.
- **Hiệu suất cao:** LWIDS được thiết kế để hoạt động hiệu quả, giảm thiểu tác động đến hiệu suất tổng thể của mạng, đặc biệt quan trọng trong các môi trường có lưu lượng truy cập cao.
- **Dễ dàng triển khai và quản lý:** LWIDS thường dễ dàng cài đặt và cấu hình, không yêu cầu thay đổi lớn về cơ sở hạ tầng mạng.
- **Phù hợp với nhiều môi trường:** LWIDS có thể được triển khai trong nhiều môi trường khác nhau, từ mạng doanh nghiệp nhỏ đến các tổ chức lớn.

Cách thức hoạt động:

LWIDS sử dụng nhiều kỹ thuật khác nhau để phát hiện các cuộc tấn công, bao gồm:

- **Phân tích dựa trên chữ ký (Signature-based detection):** So sánh lưu lượng mạng với cơ sở dữ liệu các mẫu tấn công đã biết để xác định các mối đe dọa.
- **Phân tích dựa trên bất thường (Anomaly-based detection):** Phát hiện các hành vi bất thường trong lưu lượng mạng, có thể là dấu hiệu của một cuộc tấn công mới hoặc chưa được biết đến.
- **Phân tích trạng thái giao thức (Protocol state analysis):** Kiểm tra tính hợp lệ của các gói tin theo các quy tắc giao thức để phát hiện các gói tin giả mạo hoặc không đúng định dạng.

Khi phát hiện một cuộc tấn công tiềm ẩn, LWIDS sẽ:

- **Thông báo cho quản trị viên:** Gửi cảnh báo đến quản trị viên hệ thống về hoạt động đáng ngờ, cung cấp thông tin chi tiết để họ có thể điều tra và xử lý.

So sánh với IDS truyền thống:

LWIDS khác với IDS truyền thống ở một số điểm quan trọng:

- **Hiệu suất:** LWIDS được tối ưu hóa để hoạt động hiệu quả hơn, ít ảnh hưởng đến hiệu suất mạng hơn so với IDS truyền thống.
- **Tính năng:** LWIDS có thể có ít tính năng hơn so với IDS truyền thống, nhưng vẫn cung cấp các khả năng phát hiện tấn công cơ bản.

- **Chi phí:** LWIDS thường có chi phí thấp hơn so với IDS truyền thống.

LWIDS là một giải pháp bảo mật mạng hiệu quả và tiết kiệm chi phí, phù hợp cho các tổ chức có ngân sách hạn chế hoặc yêu cầu hiệu suất cao. Mặc dù LWIDS không chủ động ngăn chặn tấn công như IPS, nhưng nó vẫn đóng vai trò quan trọng trong việc giám sát và phát hiện sớm các hoạt động đáng ngờ, giúp quản trị viên có thể phản ứng kịp thời để bảo vệ hệ thống.

1.3. Giới thiệu Snort

Snort là một hệ thống phát hiện xâm nhập mạng (Network Intrusion Detection System - NIDS) mã nguồn mở hàng đầu, được phát triển bởi Martin Roesch và hiện đang được duy trì bởi Cisco. Nó cho phép quản trị viên mạng giám sát và phân tích lưu lượng mạng thời gian thực, phát hiện các hoạt động đáng ngờ hoặc độc hại, đồng thời cung cấp cảnh báo kịp thời.

Các tính năng nổi bật của Snort:

- **Phát hiện xâm nhập dựa trên quy tắc (Rule-based detection):** Snort sử dụng một tập hợp các quy tắc (rules) để xác định các mẫu tấn công, bao gồm cả các biểu thức chính quy, giúp phát hiện các hoạt động như tấn công từ chối dịch vụ (DoS), quét cổng (port scanning), tấn công tràn bộ đệm (buffer overflow) và nhiều hơn nữa.
- **Phát hiện xâm nhập dựa trên bất thường (Anomaly-based detection):** Snort có thể học hỏi và xây dựng mô hình hoạt động mạng bình thường, từ đó phát hiện các hành vi bất thường có thể là dấu hiệu của một cuộc tấn công.
- **Linh hoạt và tùy biến:** Snort có kiến trúc module, cho phép người dùng mở rộng tính năng bằng cách thêm các plugin và preprocessor.
- **Cộng đồng lớn mạnh:** Snort có một cộng đồng người dùng và nhà phát triển tích cực, cung cấp hỗ trợ và chia sẻ các quy tắc mới để đối phó với các mối đe dọa mới nhất.
- **Miễn phí và mã nguồn mở:** Snort là một giải pháp bảo mật mạng miễn phí và mã nguồn mở, cho phép người dùng tự do kiểm tra, sửa đổi và phân phối mã nguồn.

Cách thức hoạt động:

Snort hoạt động bằng cách phân tích các gói tin mạng khi chúng đi qua một điểm giám sát trên mạng. Nó so sánh các gói tin này với các quy tắc đã được định nghĩa trước để xác định xem có bất kỳ hoạt động đáng ngờ nào hay không. Nếu phát hiện một mối đe dọa tiềm ẩn, Snort sẽ tạo ra một cảnh báo và có thể thực hiện các hành động như ghi lại thông tin gói tin, chặn kết nối, hoặc thông báo đến quản trị viên hệ thống.

Các thành phần chính của Snort:

- **Packet Decoder:** Giải mã các gói tin mạng để Snort có thể phân tích nội dung của chúng.

- **Preprocessors:** Thực hiện các tác vụ tiền xử lý trên các gói tin, chẳng hạn như giải nén, giải mã, hoặc sắp xếp lại các đoạn dữ liệu.
- **Detection Engine:** Công cụ phát hiện chính, sử dụng các quy tắc để phân tích các gói tin và xác định các mối đe dọa tiềm ẩn.
- **Logging and Alerting System:** Ghi lại thông tin về các gói tin và tạo ra cảnh báo khi phát hiện các hoạt động đáng ngờ.
- **Output Modules:** Lưu trữ dữ liệu và gửi cảnh báo đến các hệ thống khác, chẳng hạn như cơ sở dữ liệu hoặc hệ thống quản lý sự kiện bảo mật (SIEM).

Ứng dụng của Snort:

- **Phát hiện và cảnh báo về các cuộc tấn công mạng:** Snort có thể phát hiện nhiều loại tấn công khác nhau, bao gồm cả các tấn công đã biết và các tấn công mới.
- **Phân tích lưu lượng mạng:** Snort có thể được sử dụng để phân tích lưu lượng mạng để tìm ra các mẫu hoạt động bất thường hoặc đáng ngờ.
- **Ngăn chặn xâm nhập (khi được cấu hình như một NIPS):** Snort có thể được cấu hình để tự động chặn các cuộc tấn công mạng.
- **Ghi nhật ký (logging):** Snort có thể ghi lại thông tin chi tiết về các gói tin mạng, giúp quản trị viên điều tra các sự cố bảo mật.

Snort là một công cụ mạnh mẽ và linh hoạt, cung cấp một lớp bảo vệ quan trọng cho cơ sở hạ tầng mạng. Với cộng đồng hỗ trợ lớn và khả năng tùy biến cao, Snort là một lựa chọn phổ biến cho cả các tổ chức lớn và nhỏ muốn nâng cao khả năng bảo mật mạng.

CHƯƠNG 2. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

2.1. Giới thiệu

Trong thời đại mà các mối đe dọa an ninh máy tính lan tràn, việc bảo vệ hệ thống mạng chống lại các cuộc tấn công nghe lén và rình mò (man-in-the-middle) trở nên quan trọng hơn bao giờ hết. Hệ thống này giải quyết những thách thức mà các cá nhân không am hiểu công nghệ phải đối mặt khi cố gắng hiểu và sử dụng các công cụ bảo mật mạng mã nguồn mở truyền thống. Mục tiêu là tạo ra một bộ công cụ mạng có thể mở rộng, dễ tiếp cận cho tất cả người dùng, cung cấp một giải pháp nhẹ nhưng hiệu quả để phát hiện và giảm thiểu các cuộc tấn công nghe lén. Hệ thống được thiết kế chạy trên Raspberry Pi Phiên bản 3B và các model mới hơn, cũng như bất kỳ máy nào dựa trên Debian với phần cứng tương đương hoặc tốt hơn.

Thách thức trong các cuộc tấn công nghe lén:

Các cuộc tấn công nghe lén có thể có nhiều hình thức khác nhau, bao gồm cả rình mò và tấn công man-in-the-middle. Các cuộc tấn công này ngày càng trở nên tinh vi hơn theo thời gian, thường vượt xa khả năng phòng thủ của các công cụ bảo mật mạng truyền thống. Chúng tôi đề xuất một giải pháp nhẹ có thể giảm thiểu hiệu quả các cuộc tấn công như vậy với các thông báo dễ hiểu cho người dùng.

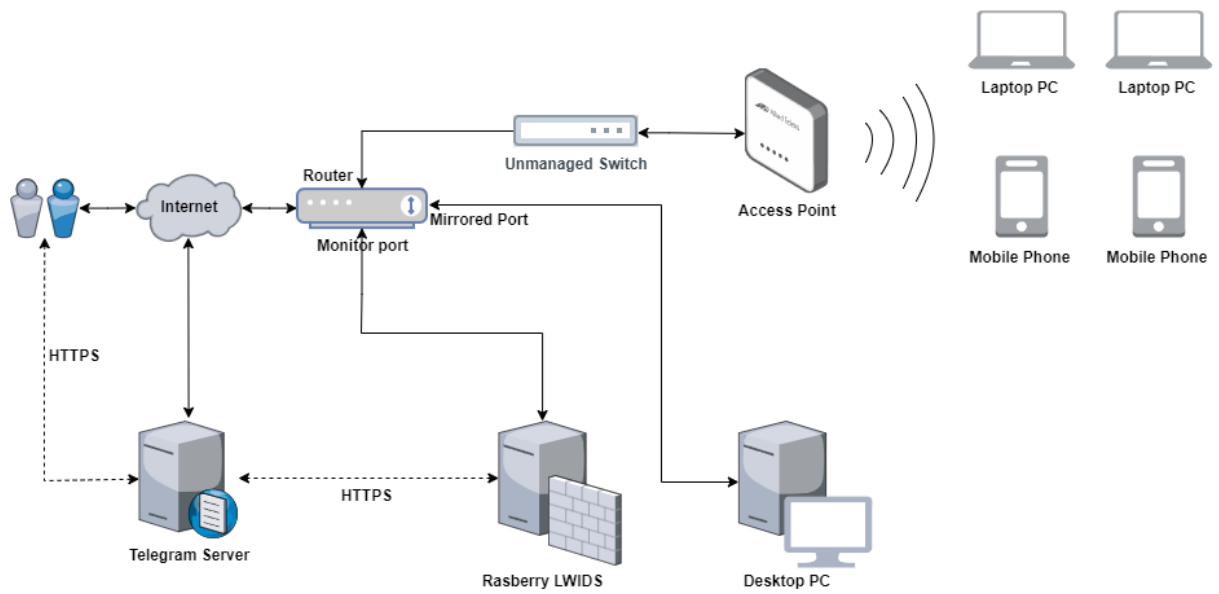
Vấn đề với phần mềm chống vi-rút và giải pháp của chúng tôi:

Phần mềm chống vi-rút, mặc dù hiệu quả chống lại vi-rút, có thể làm chậm đáng kể hiệu suất hệ thống khi chạy quét mạng. Giải pháp là chuyển khối lượng công việc này sang một máy chuyên dụng, như Raspberry Pi, hoạt động như một máy chủ DNS để giám sát lưu lượng mạng và cảnh báo người dùng về bất kỳ hoạt động truyền dữ liệu trái phép nào.

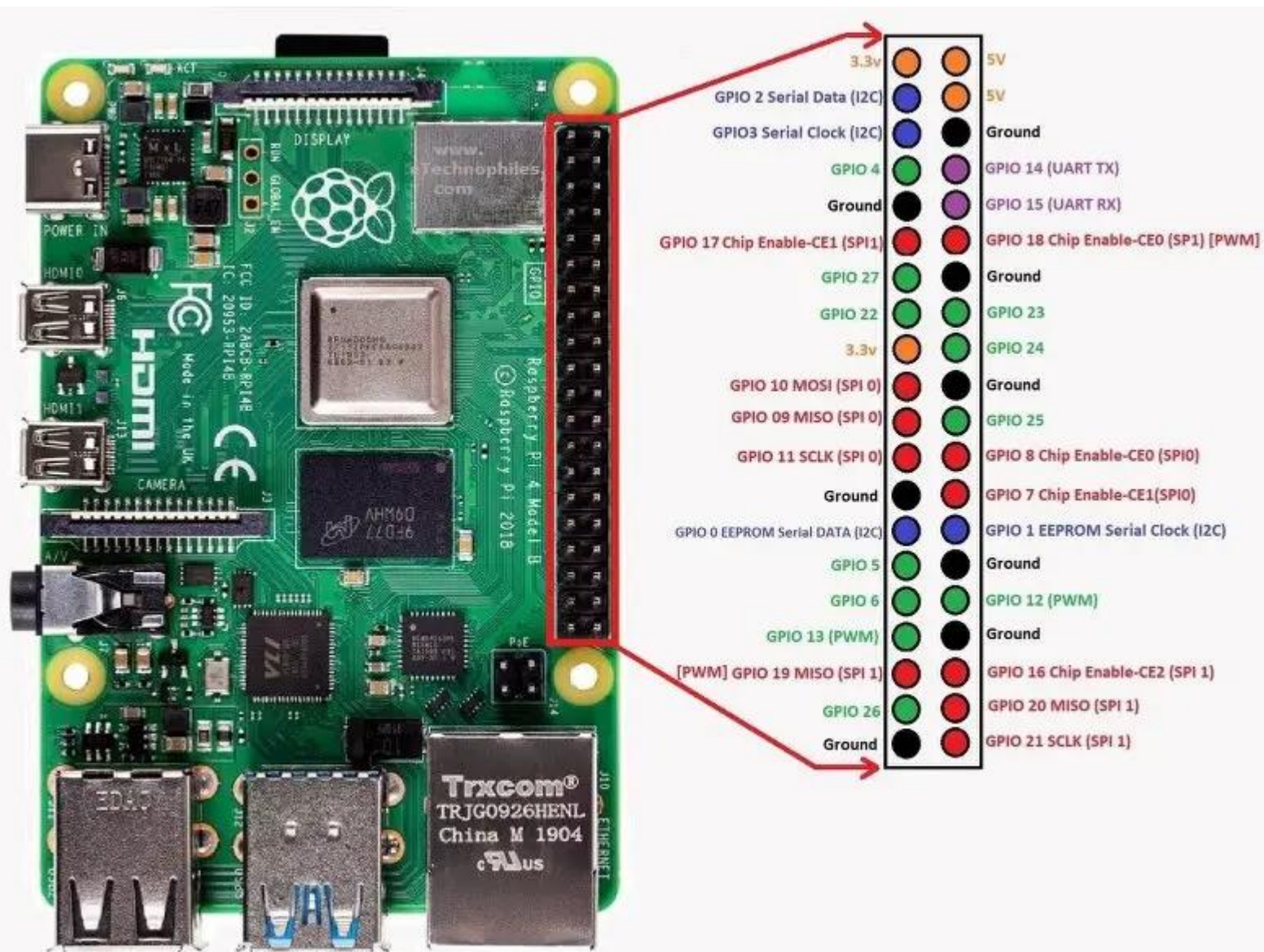
Triết lý thiết kế và khả năng tiếp cận của người dùng:

Thiết kế của hệ thống này dựa trên sự đơn giản và khả năng tiếp cận của người dùng, không phải ai cũng có chuyên môn kỹ thuật để sử dụng các công cụ bảo mật phức tạp. Hệ thống này sử dụng một giao diện đơn giản cho phép người dùng dễ dàng thiết lập và giám sát mạng. Cách tiếp cận này đảm bảo rằng tất cả người dùng có thể chủ động tham gia vào việc bảo vệ môi trường kỹ thuật số.

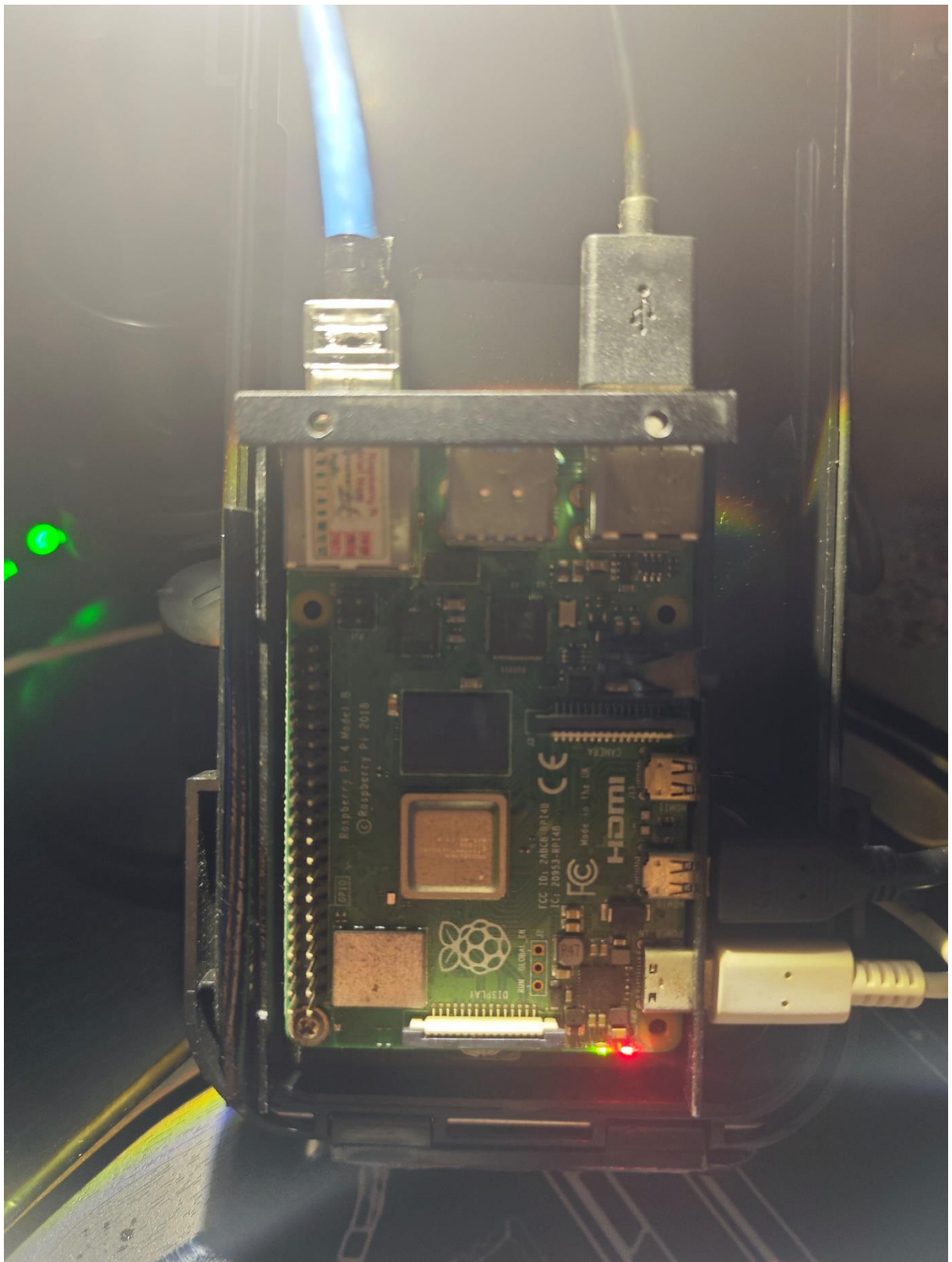
2.2. Sơ đồ thiết kế



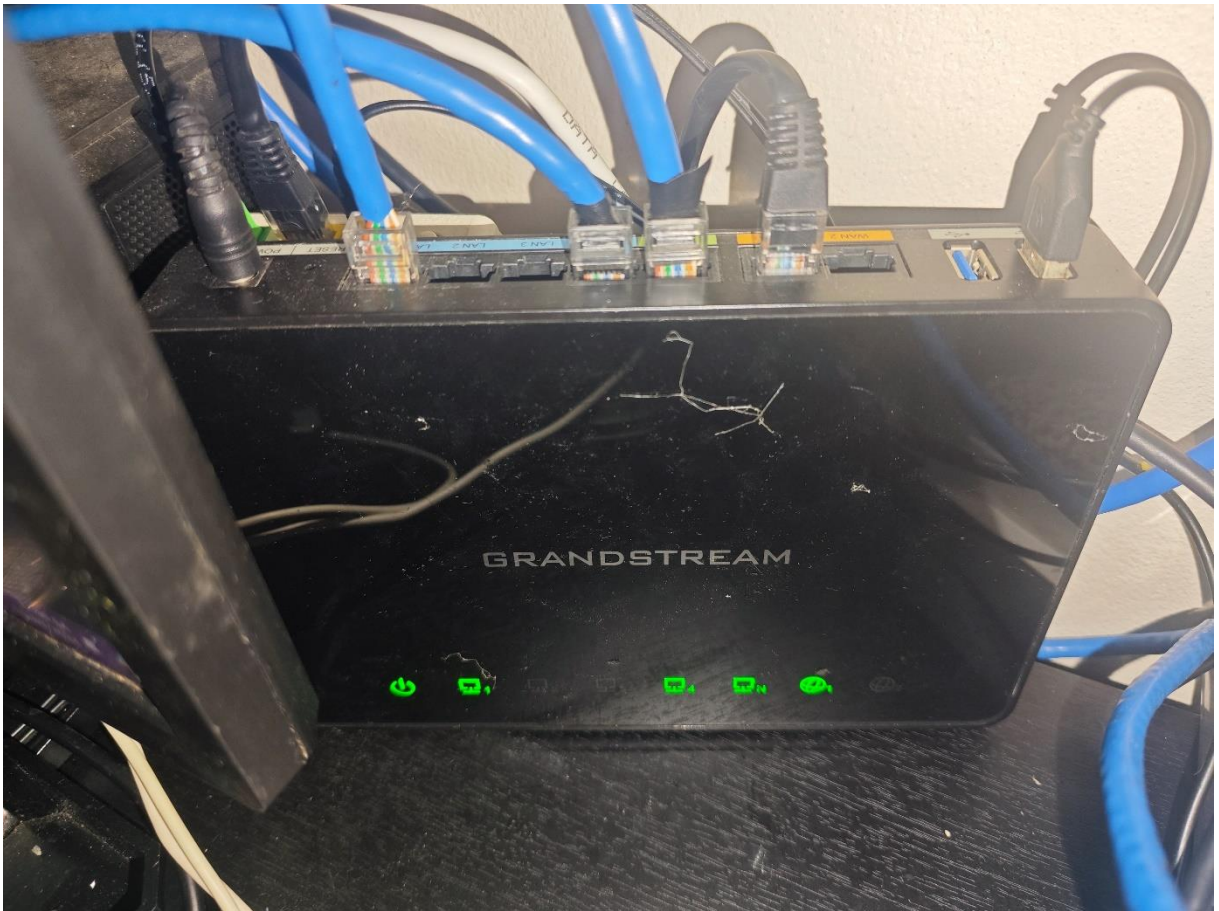
Hình 1. Mô hình mạng thực tế thử nghiệm



Hình 2. Sơ đồ chân của Raspberry Pi 4 Model B+



Hình 3. Thực tế Raspberry Pi 4 Model B+



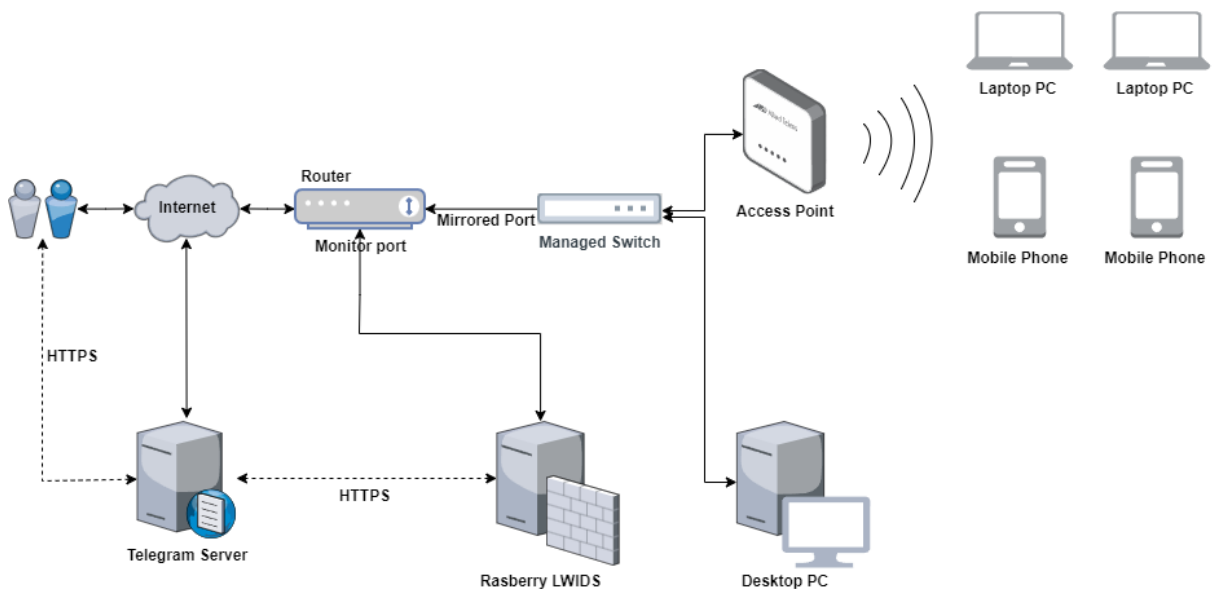
Hình 4. Router

Rasbery Pi cổng Eth0 được cắm vào cổng LAN1 là cổng mirroring
PC được cắm vào cổng LAN5 là cổng mirrored



Hình 5. Switch

Switch unmanaged được cắm vào cổng LAN4 trên router



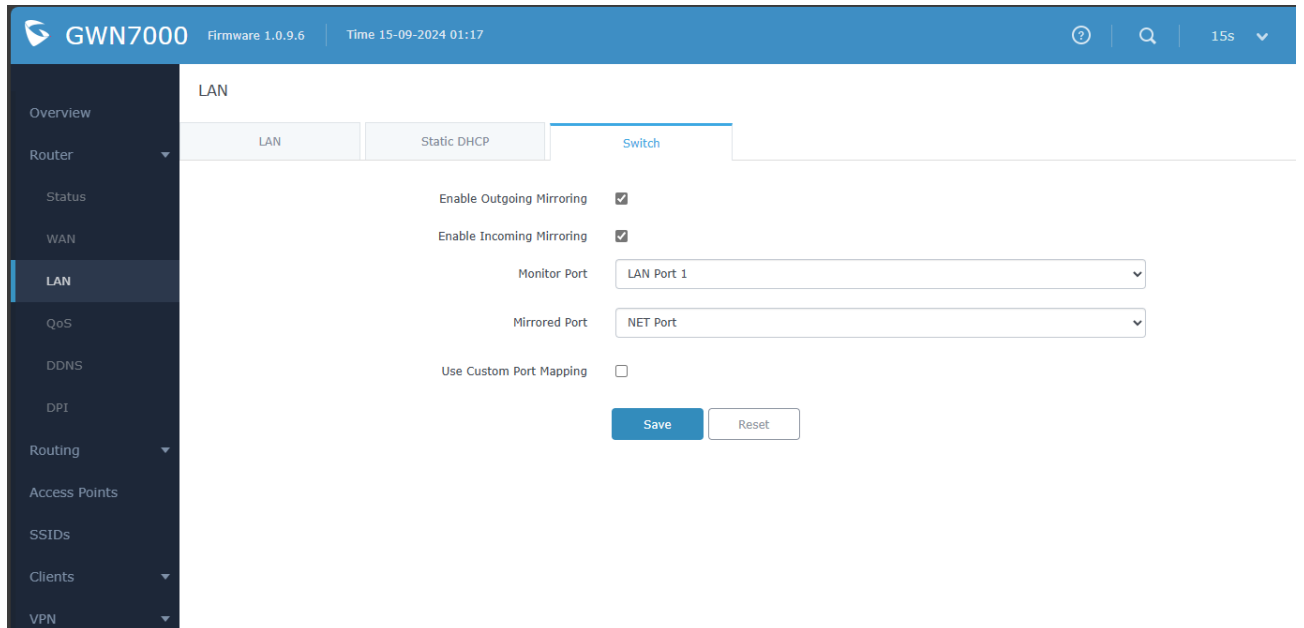
Hình 6. Mô hình mạng đề xuất sử dụng

Trong mô hình đề xuất, sử dụng managed switch để có thể cấu hình SPAN port, giúp mirror toàn bộ traffic từ các thiết bị cắm vào switch sang Raspberry Pi

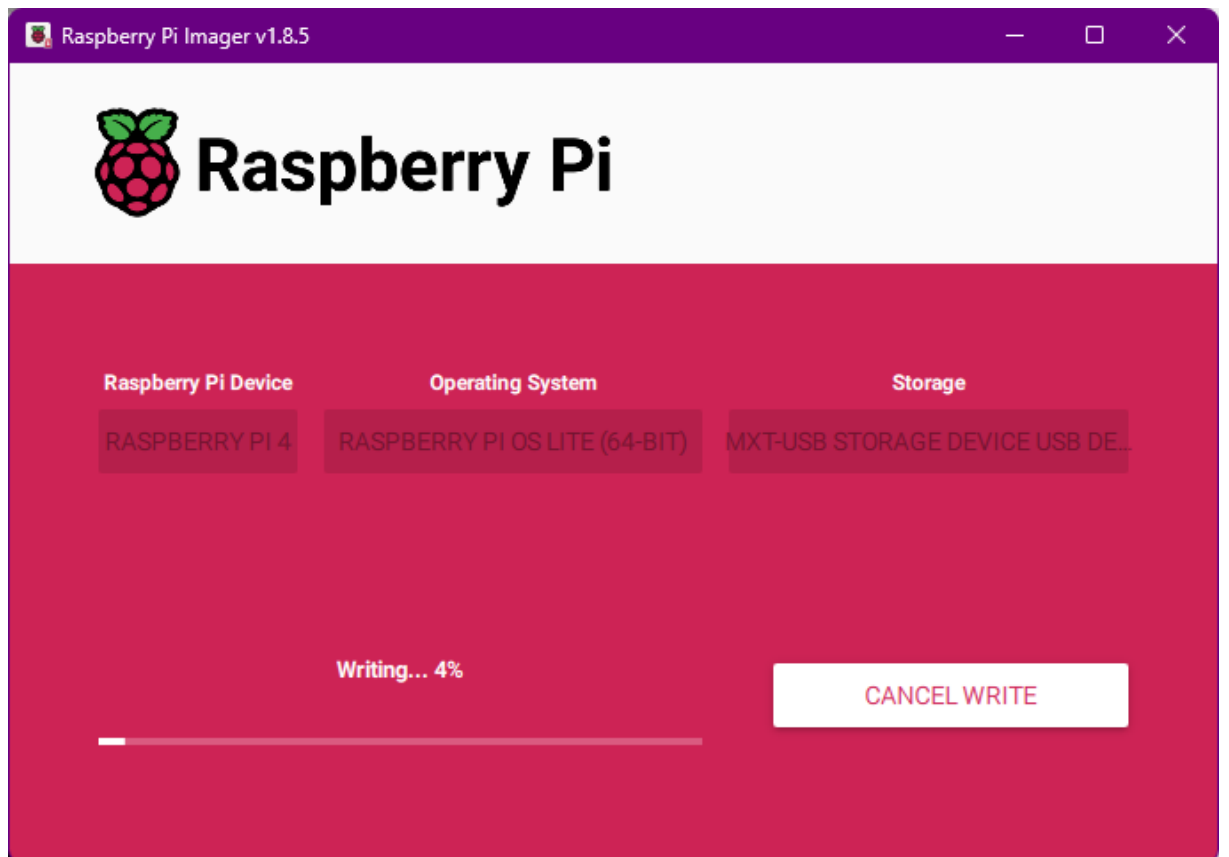
CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG

3.1. Triển khai hệ thống

Cấu hình cổng mirroring và mirrored trên router



Cài đặt bất cứ Debian-base OS nào phù hợp cho Rasberry, sử dụng Ubuntu vì Snort đã không còn hỗ trợ cho Debian Bookworm là base của RasberryOS cho Rasberry Pi



Cài đặt wireshark

```
apt install wireshark -y
```

```
root@raspberrypi:/home/longpq# apt install -y wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme at-spi2-common at-spi2-core dconf-gsettings-backend dconf-service fontconfig gsettings-desktop-schemas gstreamer1.0-gl gstreamer1.0-plugins-base gtk-update-icon-cache
hicolor-icon-theme libasynccs0 libatk-bridge2.0-0 libatk1.0-0 libatspi2.0-0 libavahi-client3 libbcg729-0 libcares2 libcairo-gobject2 libcairo2 libcdparanoia0 libcolor2 libcup2 libdatrie1
libdconf1 libdrm-amdgpu1 libdrm-nouveau2 libdrm-radeon1 libegl-mesa0 libegl1 libepoxy0 libevdev2 libfftw3-bin libfftw3-dev libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgl1
libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libgraphene-1.0-0 libgraphite2-3 libgstreamer-glib1.0-0 libgstreamer-plugins-base1.0-0 libgstreamer1.0-0 libgtk-3-0 libgtk-3-bin
libgtk-3-common libharfbuzz0b libice6 libinput-bin libinput0 liblcm2-2 liblvm2 liblua5.2-0 libmaxminddb0 libmd4c0 libminizip1 libmp3lame0 libmpeg2-0 libndev1 libopus0 liborc-0.4-0
libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpulse0 libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimedia5gsttools5
libqt5multimedia5widgets5 libqt5network5 libqt5sprintsupport5 libqt5qml5 libqt5qmlmodels5 libqt5quick5 libqt5svg5 libqt5waylandclient5 libqt5waylandcompositor5 libqt5widgets5 librsync2-2
librsync2-common libsc1 libensors-config libensors5 libsm6 libsm2ldb1 libsnappy1v5 libsndfile1 libspandsp2 libspeexdsp1 libssh-gcrypt-4 libthai-data libthai0 libtheora0 libvisual-0.4-0
libvorbisenc2 libwacom-common libwacom0 libwayland-client0 libwayland-cursor0 libwayland-egl1 libwayland-server0 libwireless-data libwireless0 libwireless1 libx11-4 libx11-xcb1
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-icccm4 libxcb-image0 libxcb-keysyms1 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-render-util0 libxcb-shape0 libxcb-shm0 libxcb-sync1
libxcb-util1 libxcb-xf86d0 libxcb-xinerama0 libxcb-xinput0 libxcb-xxkb1 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon-x11-0 libxkbcommon0 libxrandr2
libxrender1 libxshmfence1 libxtst6 libxxf86vm1 libz3-4 qt5-gtk-platformtheme qtwayland5 wireshark wireshark-common wireshark-qt x11-common
Suggested packages:
gvfs colord cups-common libvisual-0.4-plugins gstreamer1.0-tools liblcm2-utils mmd-bn opus-tools pulseaudio qgnomeplatform-qt5 qt5-image-formats-plugins qt5-qmltooling-plugins librsync2-bin
libensors snmp-mibs-downloader libwacom-bin geotupdate geop-database geop-database-extra libjs-leaflet libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
adwaita-icon-theme at-spi2-common at-spi2-core dconf-gsettings-backend dconf-service fontconfig gsettings-desktop-schemas gstreamer1.0-gl gstreamer1.0-plugins-base gtk-update-icon-cache
hicolor-icon-theme libasynccs0 libatk-bridge2.0-0 libatk1.0-0 libatspi2.0-0 libavahi-client3 libbcg729-0 libcares2 libcairo-gobject2 libcairo2 libcdparanoia0 libcolor2 libcup2 libdatrie1
libdconf1 libdrm-amdgpu1 libdrm-nouveau2 libdrm-radeon1 libegl-mesa0 libegl1 libepoxy0 libevdev2 libfftw3-bin libfftw3-dev libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgl1
libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libgraphene-1.0-0 libgraphite2-3 libgstreamer-glib1.0-0 libgstreamer-plugins-base1.0-0 libgstreamer1.0-0 libgtk-3-0 libgtk-3-bin
libgtk-3-common libharfbuzz0b libice6 libinput-bin libinput0 liblcm2-2 liblvm2 liblua5.2-0 libmaxminddb0 libmd4c0 libminizip1 libmp3lame0 libmpeg2-0 libndev1 libopus0 liborc-0.4-0
libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpulse0 libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimedia5gsttools5
libqt5multimedia5widgets5 libqt5network5 libqt5sprintsupport5 libqt5qml5 libqt5qmlmodels5 libqt5quick5 libqt5svg5 libqt5waylandclient5 libqt5waylandcompositor5 libqt5widgets5 librsync2-2
librsync2-common libsc1 libensors-config libensors5 libsm6 libsm2ldb1 libsnappy1v5 libsndfile1 libspandsp2 libspeexdsp1 libssh-gcrypt-4 libthai-data libthai0 libtheora0 libvisual-0.4-0
libvorbisenc2 libwacom-common libwacom0 libwayland-client0 libwayland-cursor0 libwayland-egl1 libwayland-server0 libwireless-data libwireless0 libwireless1 libx11-4 libx11-xcb1
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-icccm4 libxcb-image0 libxcb-keysyms1 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-render-util0 libxcb-shape0 libxcb-shm0 libxcb-sync1
libxcb-util1 libxcb-xf86d0 libxcb-xinerama0 libxcb-xinput0 libxcb-xxkb1 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon-x11-0 libxkbcommon0 libxrandr2
libxrender1 libxshmfence1 libxtst6 libxxf86vm1 libz3-4 qt5-gtk-platformtheme qtwayland5 wireshark wireshark-common wireshark-qt x11-common
0 upgraded, 152 newly installed, 0 to remove and 0 not upgraded.
Need to get 184 MB of archives.
After this operation, 474 MB of additional disk space will be used.
0% [Connecting to debian.map.fastlydns.net]
```

Hình 7. Cài đặt wireshark

Cài đặt tshark

```
apt install tshark -y
```

```
root@raspberrypi:/home/longpq# apt install -y tshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
tshark
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 153 kB of archives.
After this operation, 403 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 tshark arm64 3.6.2-2 [153 kB]
Fetched 153 kB in 2s (102 kB/s)
Selecting previously unselected package tshark.
(Reading database ... 119284 files and directories currently installed.)
Preparing to unpack .../tshark_3.6.2-2_arm64.deb ...
Unpacking tshark (3.6.2-2) ...
Setting up tshark (3.6.2-2) ...
```

Hình 8. Cài đặt tshark

Cài đặt nmap

```
apt install nmap -y
```

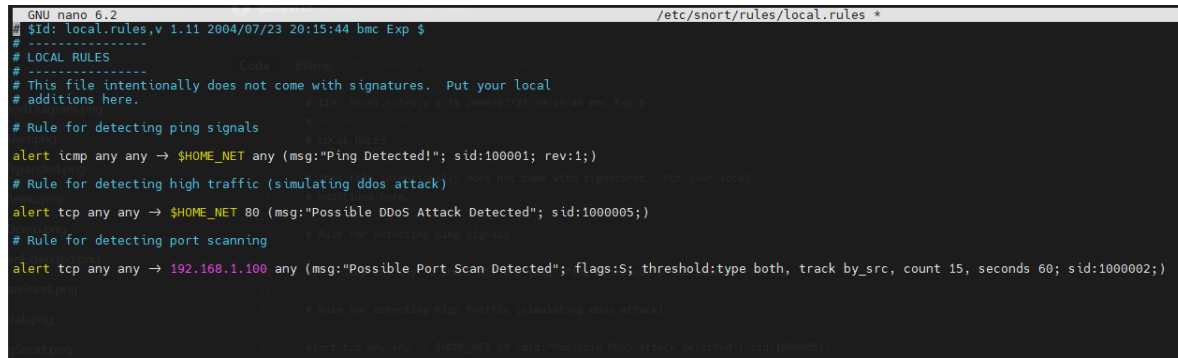

Hình 9. Cài đặt nmap

```
root@raspberrypi:/home/longps# apt install -y snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libdbmnet-libncore-locale-perl libfile-listing-perl libfont-afm-perl libhtml-form-perl libhtml-format-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-tree-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-hl-perl libio-socket-ssl-perl
  liblua5.1-5.1.2 liblua5.1-5.1-common liblua5.1-metatypes-perl liblua-perl liblua-protocol-https-perl liblua-tools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-sslayer-perl libnetfilter-queue1
  libnetmdate-perl libtiny-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl net-tools oinkmaster perl openssl-defaults snort-common snort-common-libraries snort-rules-default
Suggested packages:
  libgdmg-hmac-perl libgssapi-perl libcrypt-ssl-layer perl libsub-name-perl libbusiness-isbn-perl libauthen-ntlm-perl snort-doc
The following NEW packages will be installed:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libdbmnet-libncore-locale-perl libfile-listing-perl libfont-afm-perl libhtml-form-perl libhtml-format-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-tree-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-hl-perl libio-socket-ssl-perl
  liblua5.1-5.1.2 liblua5.1-5.1-common liblua5.1-metatypes-perl liblua-perl liblua-protocol-https-perl liblua-tools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-sslayer-perl libnetfilter-queue1
  libnetmdate-perl libtiny-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl net-tools oinkmaster perl openssl-defaults snort-common snort-common-libraries snort-rules-default
0 upgraded, 41 newly installed, 0 to remove and 2 not upgraded.
Need to get 4652 kB of archives.
After this operation, 16.1 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 liblua5.1-5.1-common all 2.1.0-beta3+dfsg-6 [44.3 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 liblua5.1-5.1-5.1 arm64 2.1.0-beta3+dfsg-6 [221 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 snort-common-libraries arm64 2.9.15-1ubuntu1 [120 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 snort-rules-default all 2.9.15-1ubuntu1 [146 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 snort-common all 2.9.15-1ubuntu1 [49.7 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy/main arm64 net-tools arm64 1.60q-2ubuntu1 [20.0 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 libdbmnet-libncore arm64 1.12-10 [27.3 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 libnetfilter-queue1 arm64 1.0.5-2 [13.5 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 libdata-dump-perl jammy/universe arm64 libdata2 arm64 2.0.7-5 [77.1 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 snort arm64 2.9.15-1ubuntu1 [777 kB]
```

Hình 10. Cài đặt snort

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your
local
# additions here.
# Rule for detecting ping signals
alert icmp any any -> $HOME_NET any (msg:"Ping Detected! Attacker
IP: $SRC IP"; sid:100001; rev:2;)
```

```
# Rule for detecting high traffic (simulating ddos attack)
alert tcp any any -> $HOME_NET 80 (msg:"Possible DDoS Attack
Detected! Attacker IP: $SRC_IP"; sid:1000005; rev:2;)
# Rule for detecting port scanning
alert tcp any any -> 192.168.1.2 any (msg:"Possible Port Scan
Detected! Attacker IP: $SRC_IP"; flags:S; threshold:type both,
track by_src, count 15, seconds 60; sid:1000002; rev:2;)
```



```
GNU nano 6.2 /etc/snort/rules/local.rules *
$Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

# Rule for detecting ping signals
alert icmp any any -> $HOME_NET any (msg:"Ping Detected!"; sid:100001; rev:1;)
# Rule for detecting high traffic (simulating ddos attack)
alert tcp any any -> $HOME_NET 80 (msg:"Possible DDoS Attack Detected"; sid:1000005;)
# Rule for detecting port scanning
alert tcp any any -> 192.168.1.100 any (msg:"Possible Port Scan Detected"; flags:S; threshold:type both, track by_src, count 15, seconds 60; sid:1000002;)
```

Hình 11. Cấu hình snort

Khởi chạy snort để lắng nghe

```
snort -q -l /var/log/snort -i eth0 -A console -c
/etc/snort/snort.conf > /home/longpq/scripts/snort_output.log
2>&1
```

3.2. Lập trình cảnh báo

Sử dụng python để gửi cảnh báo tới telegram

```
import time
import re
import requests

# Telegram bot configuration
telegram_bot_token = 'zzz'
chat_id = 'xxx'
message_thread_id = 'yyy'

# Telegram message subjects
ddos_subject = 'DDoS Attack Detected'
port_scan_subject = 'Port Scan Attack Detected'
```

```

# Path to the SNORT log file
log_file_path = '/var/log/snort/snort.alert.fast'

# Pattern to search for indicating a DDoS attack
ddos_pattern = re.compile(r"Possible DDoS Attack Detected")
port_scan_pattern = re.compile(r"Classification: Attempted
Information Leak")

def send_telegram_message(subject, message):
    url =
f'https://api.telegram.org/bot{telegram_bot_token}/sendMessage'
    data = {
        'chat_id': chat_id,
        'message_thread_id': message_thread_id,
        'text': f'{subject}: {message}'
    }
    try:
        response = requests.post(url, json=data)
        if response.status_code == 200:
            print("Telegram message sent successfully")
        else:
            print(f"Failed to send message:
{response.status_code}, {response.text}")
    except Exception as e:
        print(f"Error sending message: {e}")

def monitor_log_file():
    with open(log_file_path, 'r') as file:
        file.seek(0, 2) # Go to the end of the file
        while True:
            line = file.readline()
            if not line:
                time.sleep(0.1) # Wait briefly for new output
                continue
            if ddos_pattern.search(line):
                print(f"DDOS Alert: {line.strip()}")

```

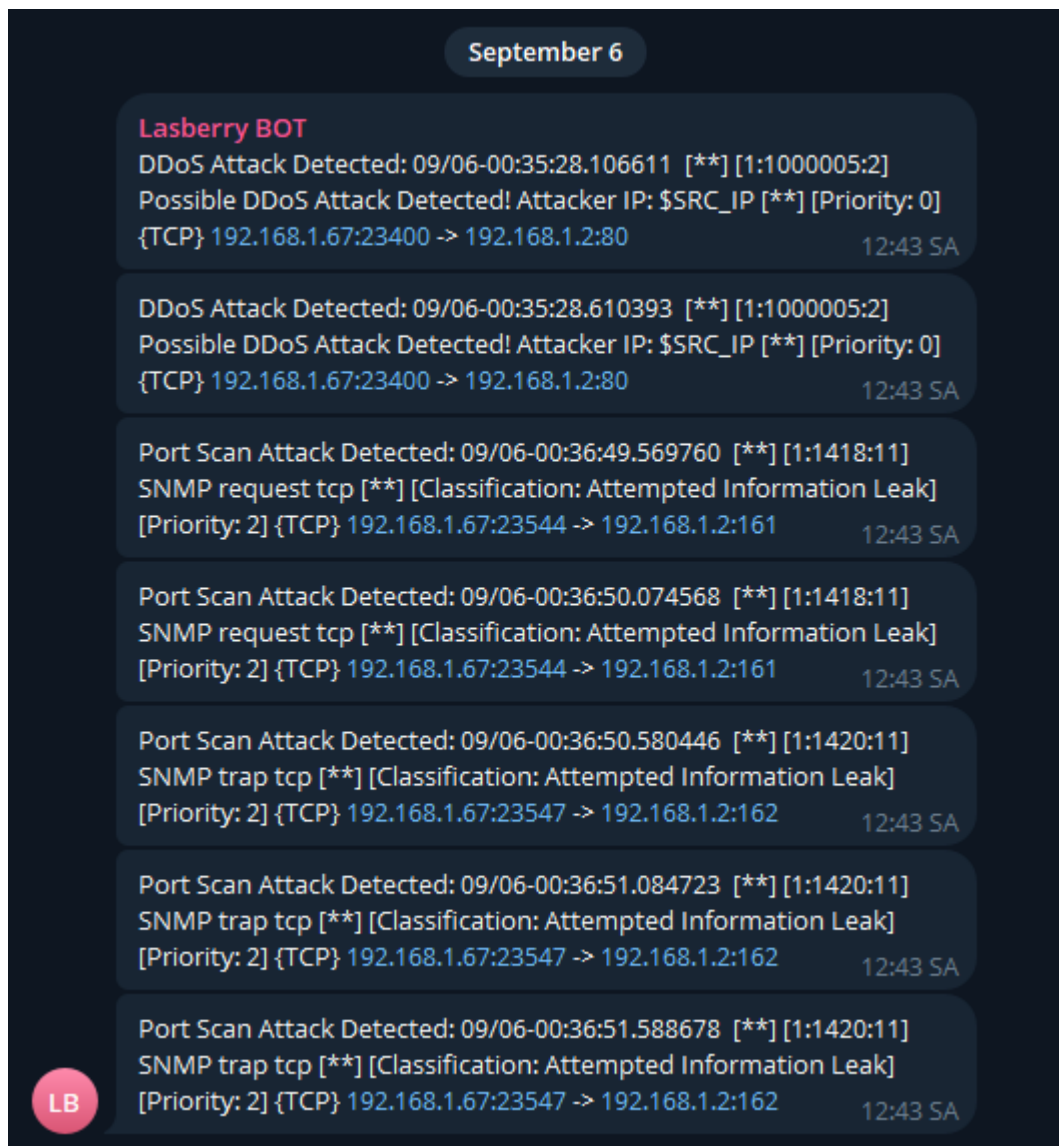
```

        send_telegram_message(ddos_subject,
line.strip())
        elif port_scan_pattern.search(line):
            print(f"Port Scan Alert: {line.strip()}")
            send_telegram_message(port_scan_subject,
line.strip())

if __name__ == "__main__":
    monitor_log_file()

```

3.3. Kết quả, đánh giá kết quả



Hình 12. Kết quả

- Snort đã có thể bắt các cuộc tấn công mạng cơ bản
- Gửi thông báo tới telegram để người quản trị nắm thông tin

Những điểm có thể cải thiện:

- Có thể cấu hình lệnh cho bot để thực hiện block các IP nghi ngờ đang thực hiện tấn công vào mạng

KẾT LUẬN

Việc triển khai Snort trên Raspberry Pi mang đến một giải pháp phát hiện xâm nhập mạng hiệu quả và tiết kiệm chi phí, đặc biệt phù hợp cho các mạng quy mô nhỏ và vừa, cũng như các dự án cá nhân. Với khả năng phân tích lưu lượng mạng thời gian thực, phát hiện các mẫu tấn công và hành vi bất thường, Snort trên Raspberry Pi cung cấp một lớp bảo vệ đáng tin cậy, giúp giám sát và cảnh báo về các mối đe dọa an ninh mạng tiềm ẩn.

Tuy nhiên, do hạn chế về tài nguyên phần cứng của Raspberry Pi, cần lưu ý đến hiệu suất khi xử lý lưu lượng mạng lớn. Ngoài ra, việc cập nhật thường xuyên các quy tắc và cấu hình Snort là cần thiết để đảm bảo hệ thống luôn hoạt động hiệu quả trước các mối đe dọa mới.

TÀI LIỆU THAM KHẢO

- [1]. <https://github.com/rajkunamaneni/TrafficDetector>
- [2]. <https://ieeexplore.ieee.org/document/7816876>
- [3]. <https://www.instructables.com/Raspberry-Pi-Firewall-and-Intrusion-Detection-Syst/>
- [4].