

MỤC LỤC

NGÀNH HỌC: AN TOÀN THÔNG TIN.....	2
HỌC PHẦN: CÁC GIAO THỨC CỦA MẠNG INTERNET	2
1. CƠ SỞ LÝ THUYẾT	2
1.1. Mô hình OSI.....	2
1.1.1. Khái niệm.....	2
1.1.2. Kiến trúc các tầng của mô hình OSI:.....	4
1.1.3. Chức năng từng tầng.....	5
1.1.4. Quá trình xử lý và vận chuyển của một gói dữ liệu trong mô hình OSI	6
1.2. Mô hình TCP/IP	10
1.3. Các công cụ chặn bắt gói tin:.....	10
1.3.1. Wireshark.....	10
1.3.2. TCP Dump	12
2. CÁC BÀI THỰC HÀNH/THÍ NGHIỆM.....	16
2.1. Bài thực hành số 1.....	16
2.1.1. Mục đích và yêu cầu	17
2.1.2. Nội dung	17
2.1.3. Ghi nhận phân tích kết quả	26
2.2. Bài thực hành số 2:.....	26
2.2.1. Mục đích và yêu cầu:.....	27
2.2.2. Nội dung:	27
2.2.3. Ghi nhận phân tích kết quả	35

NGÀNH HỌC: AN TOÀN THÔNG TIN

HỌC PHẦN: CÁC GIAO THỨC CỦA MẠNG INTERNET

BÀI THỰC HÀNH TÌM HIỂU CÁC CÔNG CỤ CHẶN BẮT GÓI TIN QUA MẠNG

1. CƠ SỞ LÝ THUYẾT

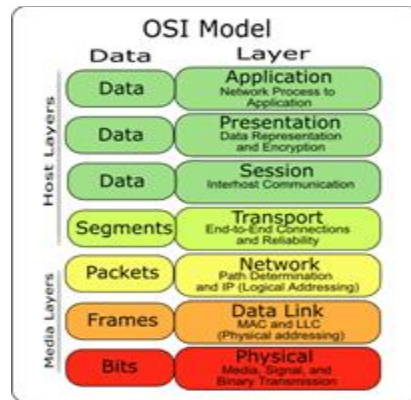
1.1. Mô hình OSI

1.1.1. Khái niệm

Mô hình OSI (*Open Systems Interconnection Reference Model*, viết tắt là *OSI Model* hoặc *OSI Reference Model*) - tạm dịch là Mô hình tham chiếu kết nối các hệ thống mở - là một thiết kế dựa vào nguyên lý tầng cấp, lý giải một cách trừu tượng kỹ thuật kết nối truyền thông giữa các máy vi tính và thiết kế giao thức mạng giữa chúng. Mô hình này được phát triển thành một phần trong kế hoạch Kết nối các hệ thống mở (*Open Systems Interconnection*) do ISO và IUT-T khởi xướng. Mô hình này được nghiên cứu phát triển từ năm 1977 và hoàn thiện vào năm 1984

Mục đích của mô hình OSI là mở rộng thông tin giữa nhiều hệ thống khác nhau mà không đòi hỏi phải có sự thay đổi về phần cứng hay phần mềm đối với hệ thống hiện hữu. Mô hình OSI không phải là giao thức (protocol) mà là mô hình giúp hiểu biết và thiết kế kiến trúc mạng một cách mềm dẻo, bền vững và dễ diễn đạt hơn.

Mô hình OSI là một khung sườn phân lớp để thiết kế mạng cho phép thông tin trong tất cả các hệ thống máy tính khác nhau. Mô hình này gồm bảy lớp riêng biệt nhưng có quan hệ với nhau, mỗi lớp nhằm định nghĩa một phân đoạn trong quá trình di chuyển thông tin qua mạng



Hình 1: Mô hình OSI

Các ưu điểm của mô hình OSI:

- Giảm độ phức tạp của truyền dữ liệu: Theo như ta biết, việc truyền dữ liệu qua lại giữa các hệ thống máy tính rất phức tạp do có rất nhiều công việc nhỏ lẻ như: dữ liệu được truyền trên cáp gì, đóng các khung dữ liệu, cách hệ thống truy nhập vào đường truyền, cơ chế truyền dẫn, xây dựng các chương trình truy nhập dữ liệu... Mô hình OSI đã phân các công việc có tính chất tương tự nhau thành từng nhóm gọi là lớp công việc, chuyên biệt hoá các lớp công việc, giúp cho các công ty sản xuất các thiết bị truyền dữ liệu có thể tập trung sản xuất các thiết bị, phần mềm phục vụ cho từng nhóm. Nhờ vậy có thể giảm sự phức tạp của truyền dữ liệu
- Chuẩn hóa các giao diện giữa các lớp: Khi xây dựng các lớp, tổ chức ISO đã quy định các tiêu chuẩn chung của các lớp và các nhà sản xuất thiết bị phục vụ truyền dữ liệu theo đó sản xuất thiết bị, dẫn tới các giao diện các lớp được chuẩn hoá.
- Chuyên môn hóa các công nghệ: Do đã được phân lớp lên các công ty có thể tập trung vào thế mạnh của mình, đầu tư tập trung công nghệ, dẫn tới các công nghệ được chuyên môn hoá cao.
- Tạo ra sự tương thích giữa các nhà sản xuất khác nhau. Nhờ có sự chuẩn hoá về giao diện nên đã tạo ra sự tương thích giữa các thiết bị của nhà sản xuất khác nhau.

- Dễ dàng trong việc dạy và học: Do đã được phân lớp nên chúng ta có thể tập trung học theo cấu trúc và tính chất của từng lớp, rất dễ dàng cho việc học tập.

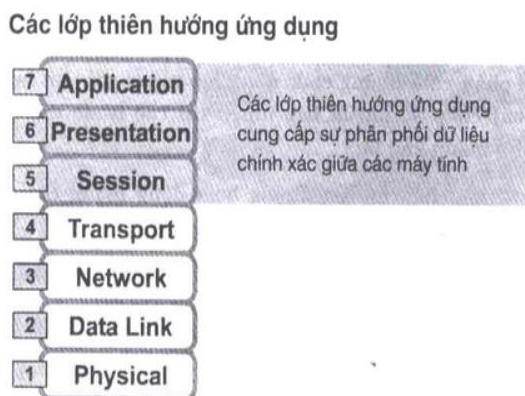
1.1.2. Kiến trúc các tầng của mô hình OSI:

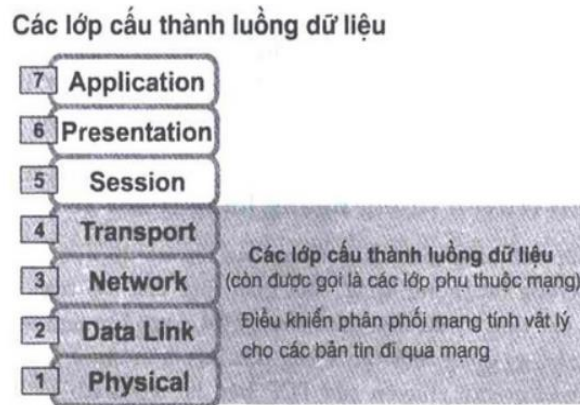
Mô hình tham chiếu OSI được chia thành bảy lớp với các chức năng cơ bản sau:

- Application (ứng dụng): giao diện giữa ứng dụng và mạng (Tầng 7)
- Presentation (trình bày): thoả thuận khuôn dạng trao đổi dữ liệu (Tầng 6)
- Session (phiên): cho phép người dùng thiết lập các kết nối (Tầng 5)
- Transport (vận chuyển): đảm bảo truyền thông giữa hai hệ thống. (Tầng 4)
- NetWork (mạng): định hướng dữ liệu truyền trong liên mạng. (Tầng 3)
- Data Link (liên kết dữ liệu): xác định việc truy xuất đến thiết bị. (Tầng 2)
- Physical (vật lý): chuyển đổi dữ liệu thành các bit và truyền đi. (Tầng 1)



Hình 2: chức năng chính của từng lớp trong mô hình OSI





Hình 3: Ba lớp phần ứng dụng và 4 lớp phần dữ liệu

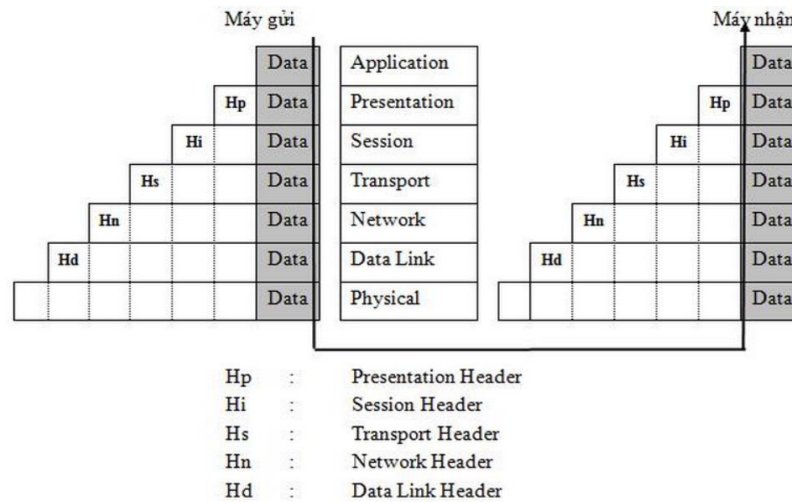
1.1.3. Chức năng từng tầng

- Tầng ứng dụng (Application layer – lớp 7): tầng ứng dụng quy định giao diện giữa người sử dụng và môi trường OSI, nó cung cấp các phương tiện cho người sử dụng truy cập và sử dụng các dịch vụ của mô hình OSI. Các ứng dụng cung được cấp như các chương trình xử lý kí tự, bảng biểu, thư tín ... và lớp 7 đưa ra các giao thức HTTP, FTP, SMTP, POP3, Telnet.
- Tầng trình bày (Presentation layer – lớp 6): chuyển đổi các thông tin từ cú pháp người sử dụng sang cú pháp để truyền dữ liệu, ngoài ra nó có thể nén dữ liệu truyền và mã hóa chúng trước khi truyền để bảo mật. Tầng này sẽ định dạng dữ liệu từ lớp 7 đưa xuống rồi gửi đi đảm bảo sao cho bên thu có thể đọc được dữ liệu của bên phát. Các chuẩn định dạng dữ liệu của lớp 6 là GIF, JPEG, PICT, MP3, MPEG ...
- Tầng giao dịch (Session layer – lớp 5): thực hiện thiết lập, duy trì và kết thúc các phiên làm việc giữa hai hệ thống. Tầng giao dịch quy định một giao diện ứng dụng cho tầng vận chuyển sử dụng. Nó xác lập ánh xạ giữa các tên đặt địa chỉ, tạo ra các tiếp xúc ban đầu giữa các máy tính khác nhau trên cơ sở các giao dịch truyền thông. Nó đặt tên nhất quán cho mọi thành phần muốn đối thoại riêng với nhau. Các giao thức trong lớp 5 sử dụng là NFS, X- Window System, ASP.

- Tầng vận chuyển (Transport layer – lớp 4): tầng vận chuyển xác định địa chỉ trên mạng, cách thức chuyển giao gói tin trên cơ sở trực tiếp giữa hai đầu nút, đảm bảo truyền dữ liệu tin cậy giữa hai đầu cuối (end-to-end). Các giao thức phổ biến tại đây là TCP, UDP, SPX.
- Tầng mạng (Network layer – lớp 3): tầng mạng có nhiệm vụ xác định việc chuyển hướng, vạch đường các gói tin trong mạng (chức năng định tuyến), các gói tin này có thể phải đi qua nhiều chặng trước khi đến được đích cuối cùng. Lớp 3 là lớp có liên quan đến các địa chỉ logic trong mạng. Các giao thức hay sử dụng ở đây là IP, RIP, IPX, OSPF, AppleTalk.
- Tầng liên kết dữ liệu (Data link layer – lớp 2): tầng liên kết dữ liệu có nhiệm vụ xác định cơ chế truy nhập thông tin trên mạng, các dạng thức chung trong các gói tin, đóng gói và phân phát các gói tin. Lớp 2 có liên quan đến địa chỉ vật lý của các thiết bị mạng, topo mạng, truy nhập mạng, các cơ chế sửa lỗi và điều khiển luồng.
- Tầng vật lý (Physical layer – lớp 1): tầng vật lý cung cấp phương thức truy cập vào đường truyền vật lý để truyền các dòng Bit không cấu trúc, ngoài ra nó cung cấp các chuẩn về điện, dây cáp, đầu nối, kỹ thuật nối mạch điện, điện áp, tốc độ cáp truyền dẫn, giao diện nối kết và các mức nối kết.

1.1.4. Quá trình xử lý và vận chuyển của một gói dữ liệu trong mô hình OSI

Việc đóng gói dữ liệu không nhất thiết xảy ra trong mỗi lần truyền dữ liệu của trình ứng dụng. Các lớp 5, 6, 7 sử dụng header trong quá trình khởi động nhưng trong phần lớn các lần truyền thì không có header của lớp 5, 6, 7 lý do là không có thông tin mới để trao đổi.



Hình 1: Quá trình xử lý và vận chuyển của một gói dữ liệu trong mô hình OSI

Các dữ liệu tại máy gửi được xử lý theo trình tự như sau:

- Người dùng thông qua lớp Application để đưa các thông tin vào máy tính. Các thông tin này có nhiều dạng khác nhau như: hình ảnh, âm thanh, văn bản, ...
- Tiếp theo các thông tin đó được chuyển xuống lớp Presentation để chuyển thành dạng chung, rồi mã hóa và nén dữ liệu.
- Tiếp đó dữ liệu được chuyển xuống lớp Session để bổ sung các thông tin về phiên giao dịch này.
- Dữ liệu tiếp tục được chuyển xuống lớp Transport, tại lớp này dữ liệu được cắt ra thành nhiều segment và bổ sung thêm các thông tin về phương thức vận chuyển dữ liệu để đảm bảo độ tin cậy khi truyền.
- Dữ liệu tiếp tục được chuyển xuống lớp Network, tại lớp này mỗi segment được cắt ra thành nhiều packet và bổ sung thêm các thông tin định tuyến.

- Tiếp đó dữ liệu được chuyển xuống lớp Data link, tại lớp này mỗi packet sẽ được cắt ra thành nhiều frame và bổ sung thêm các thông tin kiểm tra gói tin (để kiểm tra ở nơi nhận).

- Cuối cùng, mỗi frame sẽ được tầng Vật lý chuyển thành một chuỗi các bit và được đẩy lên các phương tiện truyền dẫn để truyền đến các thiết bị khác.

Quá trình truyền dữ liệu từ máy gửi đến máy nhận:

Bước 1: trình ứng dụng (trên máy gửi) tạo ra dữ liệu và các chương trình phân cứng, phần mềm cài đặt mỗi lớp sẽ bổ sung vào header và trailer (quá trình đóng gói dữ liệu tại máy gửi).

Bước 2: Lớp Physical (trên máy gửi) nhận dữ liệu.

Bước 3: Các chương trình phân cứng, phần mềm (trên máy nhận) gỡ bỏ header và trailer và xử lý phần dữ liệu (quá trình xử lý dữ liệu tại máy nhận).

Giữa bước 1 và bước 2 là quá trình tìm đường đi của gói tin. Thông thường, máy gửi đã biết địa chỉ IP của máy nhận. Vì thế sau khi xác định được địa chỉ IP của máy nhận thì lớp Network của máy gửi sẽ so sánh địa chỉ IP của máy nhận và địa chỉ IP của chính nó:

- Nếu cùng địa chỉ mạng thì máy gửi sẽ tìm đường trong bảng MAC Table của mình để có được địa chỉ MAC của máy nhận. Trong trường hợp không có được địa chỉ MAC tương ứng, nó sẽ thực hiện giao thức ARP để truy tìm địa chỉ MAC. Sau khi tìm được địa chỉ MAC, nó sẽ lưu địa chỉ MAC này vào trong bảng MAC Table để lớp Data link sử dụng ở các lần gửi sau. Sau khi có địa chỉ MAC thì máy gửi sẽ gửi gói tin đi (giao thức ARP sẽ được nói thêm trong các bài sau).

- Nếu khác địa chỉ mạng thì máy gửi sẽ kiểm tra xem máy có được khai báo Default Gateway hay không.

- Nếu có khai báo Default Gateway thì máy gửi sẽ gửi gói tin thông qua Default Gateway.
- Nếu không có khai báo Default Gateway thì máy gửi sẽ loại bỏ gói tin và thông báo "Destination host Unreachable".

Chi tiết quá trình xử lý tại máy nhận:

Bước 1: Lớp Physical kiểm tra quá trình đồng bộ bit và đặt chuỗi bit nhận được vào vùng đệm. Sau đó thông báo cho lớp Data link dữ liệu đã được nhận.

Bước 2: Lớp Data link kiểm lỗi frame bằng cách kiểm tra FCS trong trailer. Nếu có lỗi thì frame bị bỏ. Sau đó kiểm tra địa chỉ lớp Data link (địa chỉ MAC) xem có trùng với địa chỉ máy nhận hay không. Nếu đúng thì phần dữ liệu sau khi loại header và trailer sẽ được chuyển lên cho lớp Network.

Bước 3: địa chỉ lớp Network được kiểm tra xem có phải là địa chỉ máy nhận hay không (địa chỉ IP). Nếu đúng thì dữ liệu được chuyển lên cho lớp Transport xử lý.

Bước 4: Nếu giao thức lớp Transport có hỗ trợ việc phục hồi lỗi thì số định danh phân đoạn được xử lý. Các thông tin ACK, NAK (gói tin ACK, NAK dùng để phản hồi việc các gói tin đã được gửi đến máy nhận chưa) cũng được xử lý ở lớp này. Sau quá trình phục hồi lỗi và sắp thứ tự các phân đoạn, dữ liệu được đưa lên lớp Session.

Bước 5: Lớp Session đảm bảo một chuỗi các thông điệp đã trọn vẹn. Sau khi các luồng đã hoàn tất, lớp Session chuyển dữ liệu sau header lớp 5 lên cho lớp Presentation xử lý.

Bước 6: Dữ liệu được lớp Presentation xử lý bằng cách chuyển đổi dạng thức dữ liệu. Sau đó kết quả chuyển lên cho lớp Application.

Bước 7: lớp Application xử lý header cuối cùng. Header này chứa các tham số thỏa thuận giữa hai trình ứng dụng. Do vậy tham số này thường chỉ được trao đổi lúc khởi động quá trình truyền thông giữa hai trình ứng dụng.

1.2. Mô hình TCP/IP

Tham khảo lý thuyết:

Môn: Các giao thức của mạng internet

Chương 1: Tổng quan về Internet và chồng giao thức TCP/IP

1.3 Kiến trúc chồng giao thức TCP/IP

1.3. Các công cụ chặn bắt gói tin:

Hàng ngày, có hàng triệu vấn đề lỗi trong một mạng máy tính, từ việc đơn giản là nhiễm Spyware cho đến việc phức tạp như lỗi cấu hình router, và các vấn đề này không hề đơn giản cũng như không thể được xử lý tất cả lập tức. Tất cả các vấn đề trên mạng đều xuất phát ở mức gói tin, việc chặn bắt phân tích chúng là việc đầu tiên cần phải làm khi muốn xử lý các vấn đề liên quan đến mạng.

Hiện nay có khá nhiều các loại công cụ chặn bắt phân tích gói tin khác nhau như: tcpdump, Wireshark, Ettercap ... Trong bài giới thiệu này chúng ta tập trung tìm hiểu về 2 công cụ tcpdump và Wireshark.

1.3.1. Wireshark

Giới thiệu về Wireshark

Wireshark là một phần mềm mã nguồn mở dùng để bắt và phân tích các gói tin lưu thông qua card mạng của máy tính. Phần mềm này có thể sử dụng trên nhiều nền tảng khác nhau như Linux, Windows, Mac OS X, Solaris ...

Wireshark có một bề dày lịch sử. Gerald Combs là người đầu tiên phát triển phần mềm này. Phiên bản đầu tiên được gọi là Ethereal được phát hành năm 1998. Tám năm sau kể từ khi phiên bản đầu tiên ra đời, Combs từ bỏ công việc hiện tại để theo đuổi một cơ hội nghề nghiệp khác. Thật không may, tại thời điểm đó, ông không thể đạt được thỏa thuận với công ty đã thuê ông về việc bản quyền của thương hiệu Ethereal. Thay vào đó,

Combs và phần còn lại của đội phát triển đã xây dựng một thương hiệu mới cho sản phẩm “Ethereal” vào năm 2006, dự án tên là WireShark.

WireShark đã phát triển mạnh mẽ và đến nay, nhóm phát triển cho đến nay đã lên tới 500 cộng tác viên. Sản phẩm đã tồn tại dưới cái tên Ethereal không được phát triển thêm.

Lợi ích Wireshark đem lại đã giúp cho nó trở nên phổ biến như hiện nay. Nó có thể đáp ứng nhu cầu của cả các nhà phân tích chuyên nghiệp và nghiệp dư và nó đưa ra nhiều tính năng để thu hút mỗi đối tượng khác nhau.

WireShark vượt trội về khả năng hỗ trợ các giao thức (khoảng 850 loại), từ những loại phổ biến như TCP, IP đến những loại đặc biệt như là AppleTalk và Bit Torrent. Và cũng do Wireshark được phát triển trên mô hình mã nguồn mở, những giao thức mới sẽ được thêm vào. Và có thể nói rằng không có giao thức nào mà Wireshark không thể hỗ trợ.

Phần mềm Wireshark giúp:

- Người quản trị hệ thống phân tích và sửa chữa hệ thống.
- Người phát triển chương trình xây dựng các ứng dụng.
- Sinh viên tìm hiểu hoạt động của các giao thức mạng.

Các tính năng chính của Wireshark gồm:

- Bắt các gói tin đi qua một card mạng.
- Liệt kê một cách chi tiết các gói tin bắt được
- Lưu trữ và mở lại các thông tin bắt được dưới dạng file
- Tiến hành lọc các gói tin bắt được dưới nhiều tiêu chuẩn khác nhau.
- Tạo ra các biểu đồ thống kê các gói tin qua card mạng
- Và nhiều các tính năng khác

Một số ưu điểm của wireshark:

- Thân thiện với người dùng: Giao diện của Wireshark là một trong những giao diện phần mềm phân tích gói tin dễ dùng nhất. Wireshark là ứng dụng đồ họa với hệ thống menu rất rõ ràng và được bố trí dễ hiểu. Không như một số sản phẩm sử dụng dòng lệnh phức tạp như TCPdump, giao diện đồ họa của Wireshark thật tuyệt vời cho những ai đã từng nghiên cứu thế giới của phân tích giao thức.
- Giá rẻ: Wireshark là một sản phẩm miễn phí GPL. Bạn có thể tải về và sử dụng Wireshark cho bất kỳ mục đích nào, kể cả với mục đích thương mại.
- Hỗ trợ: Cộng đồng của Wireshark là một trong những cộng đồng tốt và năng động nhất của các dự án mã nguồn mở.
- Hệ điều hành hỗ trợ Wireshark: Wireshark hỗ trợ hầu hết các loại hệ điều hành hiện nay.

Hướng dẫn cài đặt và sử dụng chi tiết sẽ được giới thiệu ở phần sau

1.3.2. TCP Dump

Giới thiệu về tcpdump

Cũng như wireshark, Tcpdump là công cụ được phát triển nhằm mục đích phân tích các gói dữ liệu mạng theo dòng lệnh. Nó cho phép người dùng chặn và hiển thị các gói tin được truyền đi hoặc được nhận trên một mạng mà máy tính có tham gia.

Không thực sự có nhiều đồ họa giao diện đẹp mắt để sử dụng như Ettercap và Wireshark, tcpdump lại chỉ là một công cụ dòng lệnh với các tùy chọn được chỉ định tại thời điểm đó và cho ra các kết quả dưới dạng đầu ra chuẩn. Vì thế việc sử dụng nó sẽ khó khăn hơn với một số người dùng, nhưng nó vẫn luôn mà công cụ mạnh và linh hoạt phù hợp với nhiều nhà phân tích.

Tcpdump chủ yếu chạy trên môi trường linux hiện tại cũng đã có phiên bản cho Windows

Tcpdump xuất ra màn hình nội dung các gói tin (chạy trên card mạng mà nó đang lắng nghe) phù hợp với biểu thức logic chọn lọc mà người dùng nhập vào. Với option `-w`

người dùng có thể xuất những mô tả về gói tin này ra một file “pcap” để phân tích sau, và có thể đọc nội dung của file “pcap” đó với option -r của lệnh tcpdump, hoặc sử dụng các phần mềm khác như là: Wireshark.

Trong trường hợp không có option -c, lệnh tcpdump sẽ tiếp tục chạy cho đến khi nào nó nhận được một tín hiệu ngắt từ phía người dùng (có thể sử dụng tổ hợp phím ctrl+C hoặc sử dụng lệnh kill). Sau khi kết thúc việc bắt các gói tin, tcpdump sẽ báo cáo các cột sau:

- Packet capture: số lượng gói tin mà nó bắt được và xử lý.
- Packet received by filter: số lượng gói tin được nhận bởi bộ lọc.
- Packet dropped by kernel: số lượng packet đã bị dropped, do thiếu không gian vùng đệm, bởi cơ chế bắt gói tin của hệ điều hành.

Định dạng chung của một dòng giao thức tcpdump là:

time-stamp src > dst: flags data-seqno ack window urgent options

- Time-stamp: hiển thị thời gian gói tin được capture.
- Src và dst: hiển thị địa IP của người gửi và người nhận.
- Cờ Flag thì bao gồm các giá trị sau:
 - S(SYN): cờ này được sử dụng trong quá trình bắt tay của giao thức TCP.
 - .(ACK): cờ này được sử dụng để thông báo cho bên gửi biết là nó đã nhận được dữ liệu thành công.
 - F(FIN): được sử dụng để đóng kết nối TCP.
 - P(PUSH): thường được đặt ở cuối khối dữ liệu, đánh dấu việc truyền dữ liệu.
 - R(RST): được sử dụng khi muốn thiết lập lại đường truyền.
- Data-seqno: số sequence number của gói dữ liệu hiện tại.
- ACK: mô tả số sequence number tiếp theo của gói tin do bên gửi truyền (số sequence number mà nó mong muốn nhận được).

- Window: vùng nhớ đệm có sẵn theo hướng khác trên kết nối này.
- Urgent: cho biết có dữ liệu khẩn cấp trong gói tin.

Một số tùy chọn thông dụng trong lệnh Tcpdump:

- -i: sử dụng option này khi người dùng muốn chụp các gói tin trên một interface được chỉ định.
- -D: khi sử dụng option này, tcpdump sẽ liệt kê ra tất cả các interface đang hiện hữu trên máy tính mà nó có thể capture được.
- -c N: khi sử dụng option này, tcpdump sẽ dừng hoạt động sau khi capture N gói tin.
- -n: khi sử dụng option này, tcpdump sẽ không phân giải từ địa chỉ IP sang hostname.
- -nn: tương tự như option -n, tuy nhiên tcpdump sẽ không phân giải cả portname.
- -v: tăng số lượng thông tin về gói tin mà bạn có thể nhận được, thậm chí có thể tăng thêm với option -vv hoặc -vvv.
- -s: định nghĩa snaplength (kích thước) gói tin sẽ lưu lại, sử dụng 0 để mặc định.
- -q: khi sử dụng option này thì lệnh tcpdump sẽ hiển thị ít thông tin hơn.
- -w filename: khi sử dụng option này tcpdump sẽ capture các packet và lưu xuống file chỉ định.
- -r filename: sử dụng kèm với option -w, dùng để đọc nội dung file đã lưu từ trước.
- -x: hiển thị dữ liệu của gói tin capture dưới dạng mã Hex.
- -xx: tương tự option -x tuy nhiên sẽ chuyển đổi cả ethernet header.
- -X: hiển thị dữ liệu của gói tin capture dưới dạng mã Hex và ASCII
- -XX: tương tự như option -X tuy nhiên sẽ chuyển đổi luôn cả ethernet header.
- -A: hiển thị các packet được capture dưới dạng mã ASCII.
- -S: khi tcpdump capture packet, thì nó sẽ chuyển các số sequence number, ACK thành các relative sequence number, relative ACK. Nếu sử dụng option -S này thì nó sẽ không chuyển mà sẽ để mặc định.

- -F filename: dùng để filter các packet với các luật đã được định trước trong tập tin filename.
- -e: khi sử dụng option này, thay thì hiển thị địa chỉ IP của người gửi và người nhận, tcpdump sẽ thay thế các địa chỉ này bằng địa chỉ MAC.
- -t: khi sử dụng option này, tcpdump sẽ bỏ qua thời gian bắt được gói tin khi hiển thị cho người dùng.
- -tt: khi sử dụng option này, thời gian hiển thị trên mỗi dòng lệnh sẽ không được format theo dạng chuẩn.
- -ttt: khi sử dụng option này, thời gian hiển thị chính là thời gian chênh lệnh giữa thời gian tcpdump bắt được gói tin của gói tin và gói tin đến trước nó.
- -tttt: khi sử dụng option này, sẽ hiển thị thêm ngày vào mỗi dòng lệnh.
- -ttttt: khi sử dụng option này, thời gian hiển thị trên mỗi dòng chính là thời gian chênh lệnh giữa thời gian tcpdump bắt được gói tin của gói tin hiện tại và gói tin đầu tiên.
- -K: với option này tcpdump sẽ bỏ qua việc checksum các gói tin.
- -N: khi sử dụng option này tcpdump sẽ không in các quality domain name ra màn hình.
- -B size: sử dụng option này để cài đặt buffer_size.
- -L: hiển thị danh sách các datalink type mà interface hỗ trợ.
- -y: lựa chọn datalinktype khi bắt các gói tin.

Một số bộ lọc cơ bản:

- dst A: khi sử dụng option này, tcpdump sẽ chỉ capture các gói tin có địa chỉ đích là “A”, có thể sử dụng kèm với từ khóa net để chỉ định một dãy mạng cụ thể. Ví dụ: tcpdump dst net 192.168.1.0/24.
- src A: tương tự như option dst, nhưng thay vì capture các gói tin có địa chỉ đích cụ thể thì nó sẽ capture các gói tin có địa chỉ nguồn như quy định.

- host A: khi sử dụng option này, tcpdump sẽ chỉ capture các gói tin có địa chỉ nguồn hoặc địa chỉ đích là “A”.
- port / port range: khi sử dụng option này, tcpdump sẽ chỉ capture các gói tin có địa chỉ port được chỉ định rõ, hoặc nằm trong khoảng range định trước. Có thể sử dụng kèm với option dst hoặc src.
- less: khi sử dụng từ khóa này, tcpdump sẽ lọc (filter) các gói tin có dung lượng nhỏ hơn giá trị chỉ định.
- greater : khi sử dụng từ khóa này, tcpdump sẽ lọc (filter) các gói tin có dung lượng cao hơn giá trị chỉ định.
- (ether | ip) broadcast: capture các gói tin ip broadcast hoặc ethernet broadcast.
- (ether | ip | ip6) multicast: capture các gói tin ethernet, ip, ipv6 multicast.

Ngoài ra, tcpdump còn có thể capture các gói tin theo các protocol như: udp, tcp, icmp, ipv6 (chỉ cần gõ trực tiếp các từ khóa vào là được). Ví dụ: tcpdump icmp

Một số kết hợp trong tcpdump:

- AND : sử dụng từ khóa and hoặc &&.
- OR : sử dụng từ khóa or hoặc ||.
- EXCEPT: sử dụng từ khóa not hoặc !.

Ngoài ra để gom nhóm các điều kiện ta có thể dùng cặp từ khóa ‘’. Ví dụ: tcpdump -i eth0 ‘dst host 192.168.1.1 or 192.168.1.10 or 192.168.1.11’

2. CÁC BÀI THỰC HÀNH/THÍ NGHIỆM

2.1. Bài thực hành số 1

Tên bài: Cài đặt và thử nghiệm công cụ chặn bắt gói tin TCPDump và thư viện libcap trên môi trường linux

2.1.1. Mục đích và yêu cầu

➤ **Mục đích:**

Giúp sinh viên cài đặt công cụ chặn bắt gói tin TCPDump và thư viện libcap trên môi trường linux cụ thể ở đây là môi trường CentOS

➤ **Yêu cầu:**

- Sinh viên nắm rõ nội dung lý thuyết
- Hiểu cơ bản về mô hình OSI và TCP/IP
- Hiểu cơ bản về linux và các lệnh cơ bản trên đó

➤ **Thời gian thực hiện:**

2 tiết

➤ **Nhóm thực hiện:**

Gồm 1 sinh viên

2.1.2. Nội dung

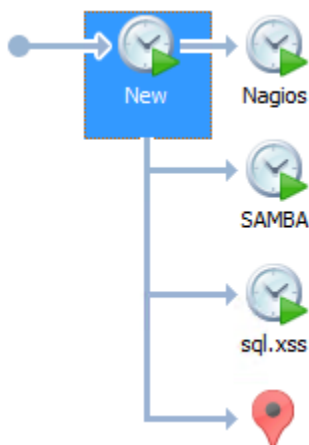
❖ **Chuẩn bị:**

Sử dụng một máy tính có kết nối mạng internet hoặc LAN để tiến hành bắt gói tin

Trên máy tính cài đặt CentOS 6.5 (Có thể cài đặt trực tiếp hoặc ảo hóa qua vmware)

❖ **Các bước thực hiện:**

Chạy snapshot trên CentOS



Bước 1: Cài đặt libpcap

Trước khi muốn cài đặt TCPDump chúng ta cần cài đặt libpcap trước vì đây là thư viện chuẩn hỗ trợ việc bắt và đọc nội dung gói tin.

Có 2 cách để cài đặt libpcap trên CentOS

➤ Cách 1:

- Download bộ cài libpcap về ở đây chúng ta sẽ dùng bản libpcap-1.6.1:

Lên trang <http://www.tcpdump.org/> xem thông tin và download bản libpcap mới nhất. Ở trong bài lab này sử dụng libpcap-1.6.1.tar.gz

LATEST RELEASE

Version: 4.6.1 / 1.6.1

Release Date: Jul 19, 2014

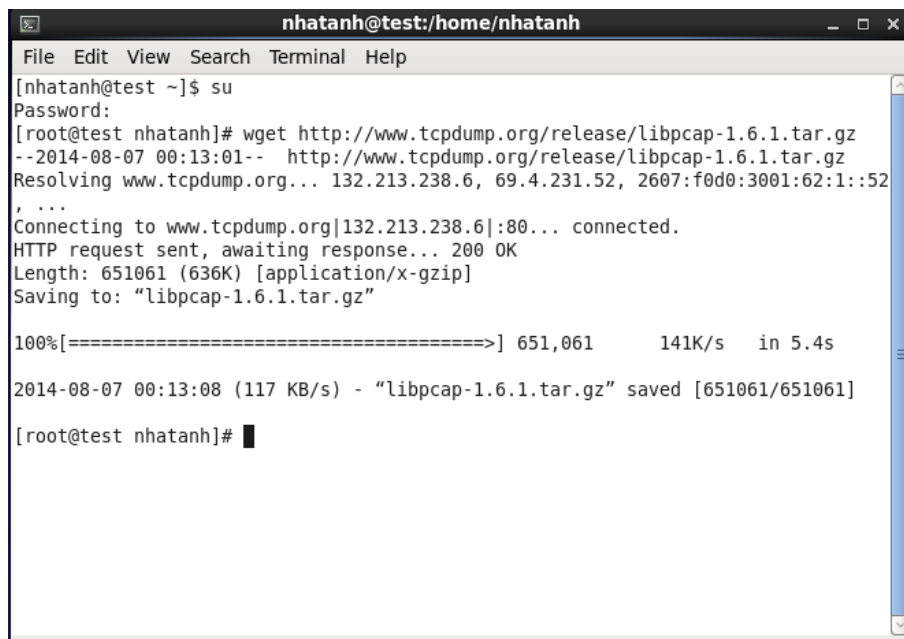
Version 4.6.0/1.6.0 revised for minor fixes discovered during rc process

- [tcpdump-4.6.1.tar.gz](#) (changelog) (PGP signature)
- [libpcap-1.6.1.tar.gz](#) (changelog) (PGP signature)
- [tcpdump-workers.asc](#) (NEW! tcpdump.org signing key)

Hình 2: Download libpcap trên trang www.tcpdump.org

Hoặc vào Terminal download qua bằng lệnh:

#wget <http://www.tcpdump.org/release/libpcap-1.6.1.tar.gz>



```
nhatanh@test:/home/nhatanh
File Edit View Search Terminal Help
[nhatanh@test ~]$ su
Password:
[root@test nhatanh]# wget http://www.tcpdump.org/release/libpcap-1.6.1.tar.gz
--2014-08-07 00:13:01-- http://www.tcpdump.org/release/libpcap-1.6.1.tar.gz
Resolving www.tcpdump.org... 132.213.238.6, 69.4.231.52, 2607:f0d0:3001:62:1::52
...
Connecting to www.tcpdump.org|132.213.238.6|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 651061 (636K) [application/x-gzip]
Saving to: "libpcap-1.6.1.tar.gz"

100%[=====] 651,061      141K/s   in 5.4s

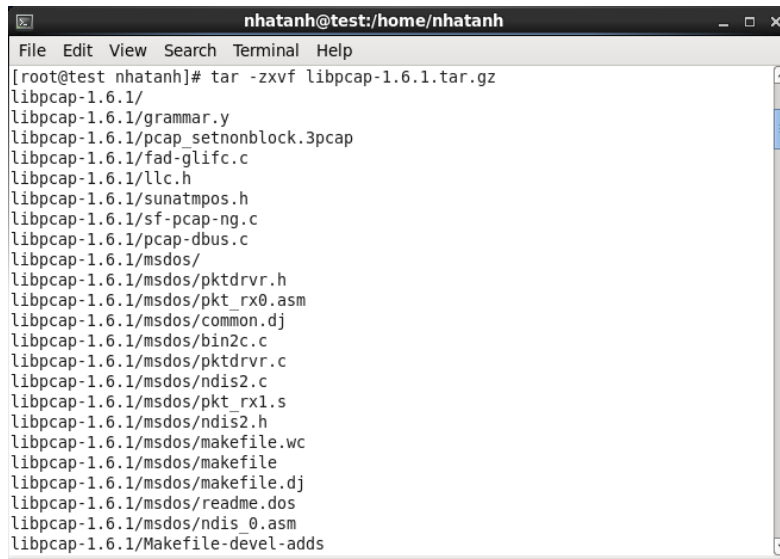
2014-08-07 00:13:08 (117 KB/s) - "libpcap-1.6.1.tar.gz" saved [651061/651061]

[root@test nhatanh]#
```

Hình 3: Sử dụng lệnh wget để download thư viện libpcap

- Giải nén file download về, vì nó có đuôi tar.gz nên cần dùng lệnh sau:

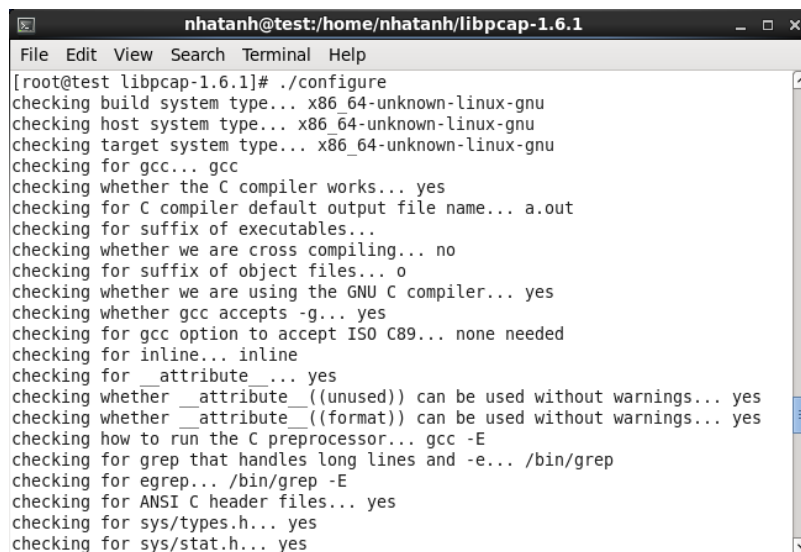
#tar -zxvf libpcap-1.6.1.tar.gz



```
nhatanh@test:/home/nhatanh
File Edit View Search Terminal Help
[root@test nhatanh]# tar -zxvf libpcap-1.6.1.tar.gz
libpcap-1.6.1/
libpcap-1.6.1/grammar.y
libpcap-1.6.1/pcap_setnonblock.3pcap
libpcap-1.6.1/fad_glibc.c
libpcap-1.6.1/llc.h
libpcap-1.6.1/sunatmpos.h
libpcap-1.6.1/sf-pcap-ng.c
libpcap-1.6.1/pcap-dbus.c
libpcap-1.6.1/msdos/
libpcap-1.6.1/msdos/pktdrvr.h
libpcap-1.6.1/msdos/pkt_rx0.asm
libpcap-1.6.1/msdos/common.dj
libpcap-1.6.1/msdos/bin2c.c
libpcap-1.6.1/msdos/pktdrvr.c
libpcap-1.6.1/msdos/ndis2.c
libpcap-1.6.1/msdos/pkt_rx1.s
libpcap-1.6.1/msdos/ndis2.h
libpcap-1.6.1/msdos/makefile.wc
libpcap-1.6.1/msdos/makefile
libpcap-1.6.1/msdos/makefile.dj
libpcap-1.6.1/msdos/readme.dos
libpcap-1.6.1/msdos/ndis_0.asm
libpcap-1.6.1/Makefile-devel-adds
```

Hình 4: Giải nén file libpcap-1.6.1.tar.gz

- Vào thư mục libpcap-1.6.1 bằng lệnh: **#cd libpcap-1.6.1**
- Khi đã vào được thư mục libpcap tiếp theo cần tiến hành Configure libpcap bằng lệnh: **#./configure**



```
nhatanh@test:/home/nhatanh/libpcap-1.6.1
File Edit View Search Terminal Help
[root@test libpcap-1.6.1]# ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for inline... inline
checking for __attribute__... yes
checking whether __attribute__((unused)) can be used without warnings... yes
checking whether __attribute__((format)) can be used without warnings... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
```

Hình 5: Configure libpcap bằng lệnh ./configure

- Compile libpcap trên CentOS bằng lệnh: **#make**

```
nhatanh@test:/home/nhatanh/libpcap-1.6.1
File Edit View Search Terminal Help
[root@test libpcap-1.6.1]# make
bison -y -p pcap_ -d grammar.y
conflicts: 38 shift/reduce
mv y.tab.c grammar.c
mv y.tab.h tokdefs.h
gcc -fpic -I. -DHAVE_CONFIG_H -D_U="__attribute__((unused))" -g -O2 -c scanner.c
gcc -fpic -I. -DHAVE_CONFIG_H -D_U="__attribute__((unused))" -g -O2 -DYYLVAL=
pcap_lval -c grammar.c
rm -f bpf_filter.c
ln -s ./bpf/net/bpf_filter.c bpf_filter.c
gcc -fpic -I. -DHAVE_CONFIG_H -D_U="__attribute__((unused))" -g -O2 -c bpf_fi
lter.c
if grep GIT ./VERSION >/dev/null; then \
    read ver <./VERSION; \
    echo $ver | tr -d '\012'; \
    date +%Y_%m_%d; \
else \
    cat ./VERSION; \
fi | sed -e 's/./char pcap_version[] = "&"/' > version.c
gcc -fpic -I. -DHAVE_CONFIG_H -D_U="__attribute__((unused))" -g -O2 -c versio
n.c
ar rc libpcap.a pcap-linux.o pcap-usb-linux.o pcap-can-linux.o pcap-netfilter-li
nux.o fad-getad.o pcap.o inet.o gencode.o optimize.o nametoaddr.o etherent.o sav
efile.o sf-pcap.o sf-pcap-ng.o pcap-common.o bpf_image.o bpf_dump.o scanner.o g
```

Hình 6: Compile libpcap trên CentOS bằng lệnh make

- Cuối cùng install libpcap bằng lệnh **#make install**

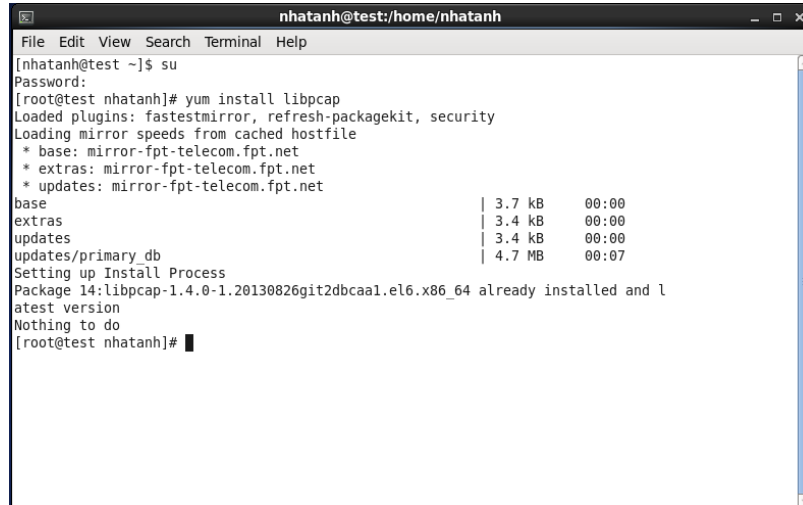
```
nhatanh@test:/home/nhatanh/libpcap-1.6.1
File Edit View Search Terminal Help
[root@test libpcap-1.6.1]# make install
VER=`cat ./VERSION`; \
    MAJOR_VER=`sed 's/\([0-9]\)[0-9]*\)\.*/\1/' ./VERSION`; \
    gcc -shared -Wl,-soname,libpcap.so.$MAJOR_VER \
    -o libpcap.so.$VER pcap-linux.o pcap-usb-linux.o pcap-can-linux.o pcap-n
etfilter-linux.o fad-getad.o pcap.o inet.o gencode.o optimize.o nametoaddr.o etheren
t.o savefile.o sf-pcap.o sf-pcap-ng.o pcap-common.o bpf_image.o bpf_dump.o scanner.
o grammar.o bpf_filter.o version.o
[ -d /usr/local/lib ] || \
    (mkdir -p /usr/local/lib; chmod 755 /usr/local/lib)
VER=`cat ./VERSION`; \
    MAJOR_VER=`sed 's/\([0-9]\)[0-9]*\)\.*/\1/' ./VERSION`; \
    /usr/bin/install -c libpcap.so.$VER /usr/local/lib/libpcap.so.$VER; \
    ln -sf libpcap.so.$VER /usr/local/lib/libpcap.so.$MAJOR_VER; \
    ln -sf libpcap.so.$MAJOR_VER /usr/local/lib/libpcap.so
#
# Most platforms have separate suffixes for shared and
# archive libraries, so we install both.
#
[ -d /usr/local/lib ] || \
    (mkdir -p /usr/local/lib; chmod 755 /usr/local/lib)
/usr/bin/install -c -m 644 libpcap.a /usr/local/lib/libpcap.a
ranlib /usr/local/lib/libpcap.a
[ -d /usr/local/lib ] || \
    (mkdir -p /usr/local/lib; chmod 755 /usr/local/lib)
```

Hình 7: install libpcap bằng lệnh make install

- Cách 2:

CentOS hỗ trợ lệnh install chúng ta có thể download và cài đặt chỉ bằng một lệnh

#yum install libpcap



```
nhatanh@test:/home/nhatanh
File Edit View Search Terminal Help
[nhatanh@test ~]$ su
Password:
[root@test nhatanh]# yum install libpcap
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
* base: mirror-fpt-telecom.fpt.net
* extras: mirror-fpt-telecom.fpt.net
* updates: mirror-fpt-telecom.fpt.net
base                                     | 3.7 kB    00:00
extras                                 | 3.4 kB    00:00
updates                               | 3.4 kB    00:00
updates/primary.db                    | 4.7 MB    00:07
Setting up Install Process
Package 14:libpcap-1.4.0-1.20130826git2dbca1.el6.x86_64 already installed and l
atest version
Nothing to do
[root@test nhatanh]#
```

Hình 8: download libcap bằng lệnh yum

Bước 2: cài đặt TCPDump

Sau khi cài đặt thành công Libpcap tiến hành cài đặt TCPDump. Tương tự như libpcap cũng sẽ có 2 cách để cài đặt TCPDump

➤ Cách 1:

Cũng tương tự như với libpcap chúng ta có thể download phiên bản mới nhất của TCPDump tại <http://www.tcpdump.org/>. Ở đây sử dụng bản tcpdump-4.6.1

Tiến hành cài đặt tương tự bằng các lệnh sau:

```
# wget http://www.tcpdump.org/release/tcpdump-4.6.1.tar.gz
```

```
# tar -xzf tcpdump-4.6.1.tar.gz
```

```
# cd tcpdump-4.6.1
```

```
# ./configure
```

```
# make
```

make install

```

nhatanh@test:/home/nhatanh/tcpdump-4.6.1
File Edit View Search Terminal Help
[root@test tcpdump-4.6.1]# make install
[ -d /usr/local/sbin ] || \
    (mkdir -p /usr/local/sbin; chmod 755 /usr/local/sbin)
/usr/bin/install -c tcpdump /usr/local/sbin/tcpdump
/usr/bin/install -c tcpdump /usr/local/sbin/tcpdump.`cat ./VERSION`
[ -d /usr/local/share/man/man1 ] || \
    (mkdir -p /usr/local/share/man/man1; chmod 755 /usr/local/share/man/
man1)
/usr/bin/install -c -m 644 tcpdump.1 /usr/local/share/man/man1/tcpdump.1
[root@test tcpdump-4.6.1]# tcpdump -i eth0 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
00:57:14.962893 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19
2.168.48.132.40814: Flags [FP.], seq 758815975, ack 735183108, win 64240, length
0
00:57:15.066121 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19
2.168.48.132.40814: Flags [FP.], seq 0, ack 1, win 64240, length 0
00:57:15.166118 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19
2.168.48.132.40814: Flags [FP.], seq 0, ack 1, win 64240, length 0
00:57:15.266352 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19
2.168.48.132.40814: Flags [FP.], seq 0, ack 1, win 64240, length 0
00:57:15.366333 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19
2.168.48.132.40814: Flags [FP.], seq 0, ack 1, win 64240, length 0
00:57:15.466456 IP a23-201-102-58.deploy.static.akamaitechnologies.com.http > 19

```

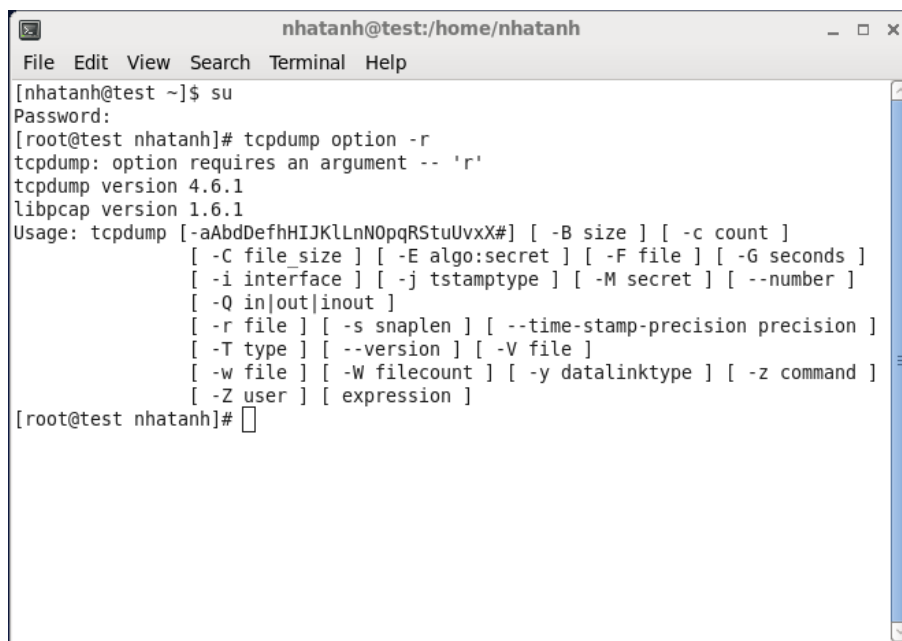
Hình 9: make install thành công tcpdump

➤ Cách 2:

Sử dụng lệnh yum install tcpdump để download và cài đặt TCPDump

Bước 3: Thử nghiệm sử dụng công cụ TCPDump

Sau khi cài đặt thành công libpcap và TCPDump. Kiểm tra option TCPDump bằng câu lệnh tcpdump option -r



```

nhatanh@test:/home/nhatanh
File Edit View Search Terminal Help
[nhatanh@test ~]$ su
Password:
[root@test nhatanh]# tcpdump option -r
tcpdump: option requires an argument -- 'r'
tcpdump version 4.6.1
libpcap version 1.6.1
Usage: tcpdump [-aAbDDefhHIJKlLnNOpqRStuUvxxX#] [-B size] [-c count]
               [-C file_size] [-E algo:secret] [-F file] [-G seconds]
               [-i interface] [-j tstamptype] [-M secret] [--number]
               [-Q in|out|inout]
               [-r file] [-s snaplen] [--time-stamp-precision precision]
               [-T type] [--version] [-V file]
               [-w file] [-W filecount] [-y datalinktype] [-z command]
               [-Z user] [expression]
[root@test nhatanh]#

```

Hình 10: Các option của tcpdump

Ở hình vẽ trên chúng ta thấy version của tcpdump và libpcap là 4.6.1 và 1.6.1 giống như đã cài đặt ở phần trên

Như đã giới thiệu ở trên TCPDump hỗ trợ khá nhiều option khác nhau giúp người sử dụng thuận lợi trong việc lọc phân tích các gói tin theo yêu cầu một cách hiệu quả.

Sử dụng lệnh **#tcpdump -D** để xem các interface


```

nhatanh@test:/home/nhatanh
File Edit View Search Terminal Help
[nhatanh@test ~]$ su
Password:
[root@test nhatanh]# tcpdump -D
1.eth0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.nflog (Linux netfilter log (NFLOG) interface)
5.nfqueue (Linux netfilter queue (NFQUEUE) interface)
6.usbmon1 (USB bus number 1)
7.usbmon2 (USB bus number 2)
[root@test nhatanh]#

```

Hình 11: xem interface qua lệnh `tcpdump -D`

Bắt các gói tin Ping ICMP trên card mạng eth0 bằng lệnh `#tcpdump -i eth0 icmp`

```

ATT@ATT:~
File Edit View Search Terminal Help
[root@ATT ~]# tcpdump -i eth0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:55:00.084848 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1
68, length 40
10:55:00.084899 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 168
, length 40
10:55:01.092352 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1
69, length 40
10:55:01.092378 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 169
, length 40
10:55:02.107956 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1
70, length 40
10:55:02.107981 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 170
, length 40
10:55:03.123630 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1
71, length 40
10:55:03.123658 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 171
, length 40
10:55:04.139316 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1
72, length 40
10:55:04.139342 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 172
, length 40

```

Hình 12: Bắt gói tin Ping ICMP bằng lệnh `tcpdump -i eth0 icmp`

2.1.3. Ghi nhận phân tích kết quả

❖ Kết quả mong muốn

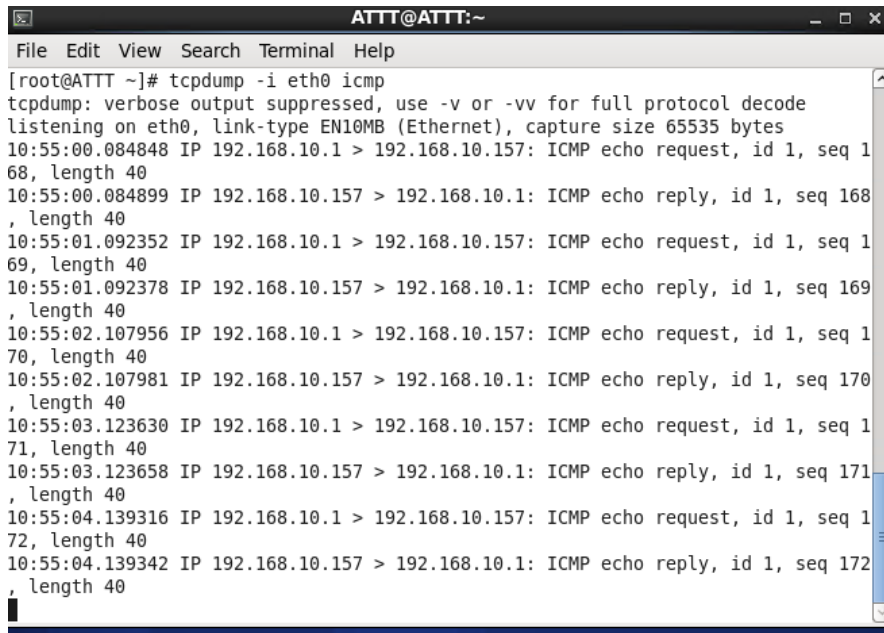
Cài đặt thành công công cụ chặn bắt gói tin TCPDump và thư viện libpcap trên môi trường linux

Sử dụng được một số chức năng của tcpdump

Hiểu được cơ chế làm việc của libpcap và TCPDump

❖ Kết quả thực hiện

Sau khi cài đặt thành công tcpdump tiến hành chạy tcpdump kiểm tra các card mạng và thử bắt các gói tin Ping ICMP trên card mạng eth0



```
ATT@ATT:~  
File Edit View Search Terminal Help  
[root@ATT ~]# tcpdump -i eth0 icmp  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes  
10:55:00.084848 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1  
68, length 40  
10:55:00.084899 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 168  
, length 40  
10:55:01.092352 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1  
69, length 40  
10:55:01.092378 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 169  
, length 40  
10:55:02.107956 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1  
70, length 40  
10:55:02.107981 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 170  
, length 40  
10:55:03.123630 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1  
71, length 40  
10:55:03.123658 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 171  
, length 40  
10:55:04.139316 IP 192.168.10.1 > 192.168.10.157: ICMP echo request, id 1, seq 1  
72, length 40  
10:55:04.139342 IP 192.168.10.157 > 192.168.10.1: ICMP echo reply, id 1, seq 172  
, length 40
```

2.2. Bài thực hành số 2:

Tên bài: Cài đặt và sử dụng công cụ chặn bắt gói tin Wireshark và thư viện winpcap trên môi trường Windows

2.2.1. Mục đích và yêu cầu:

➤ Mục đích:

Giúp sinh viên cài đặt công cụ chặn bắt gói tin wireshark và thư viện winpcap trên môi trường Windows cụ thể ở đây là Windows 7

➤ Yêu cầu:

- Sinh viên nắm rõ nội dung lý thuyết
- Hiểu cơ bản về mô hình OSI và TCP/IP

➤ Thời gian thực hiện:

2 tiết

➤ Nhóm thực hiện:

Gồm 1 sinh viên

2.2.2. Nội dung:

❖ Chuẩn bị:

Sử dụng một máy tính có kết nối mạng internet hoặc LAN để tiến hành bắt gói tin

Trên máy tính cài đặt hệ điều hành Windows 7 (Hoặc có thể là một hệ điều hành Windows khác)

❖ Các bước thực hiện:

Bước 1: Cài đặt winpcap

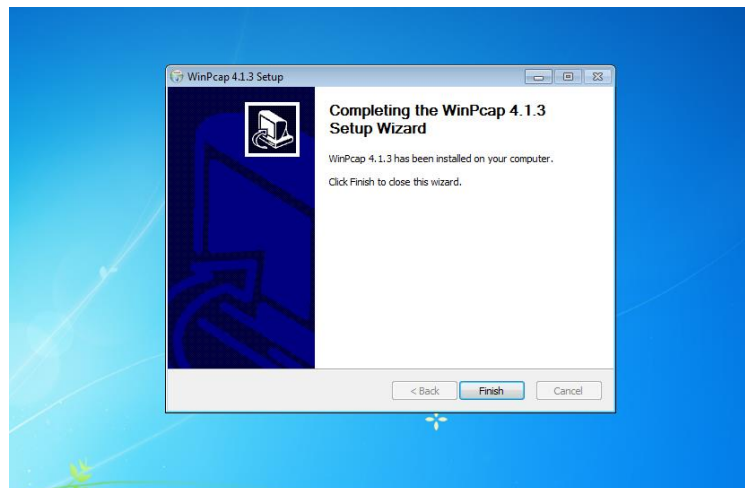
Cũng như tcpdump và winpcap, wireshark muốn hoạt động được cần có thư viện winpcap hỗ trợ việc chặn bắt và xem nội dung gói tin



Download winpcap bản mới nhất trên trang <http://www.winpcap.org/>. Ở trong bài này chúng ta sẽ dùng bản winpcap 4.1.3

Hình 13: Download winpcap 4.1.3 cho Windows

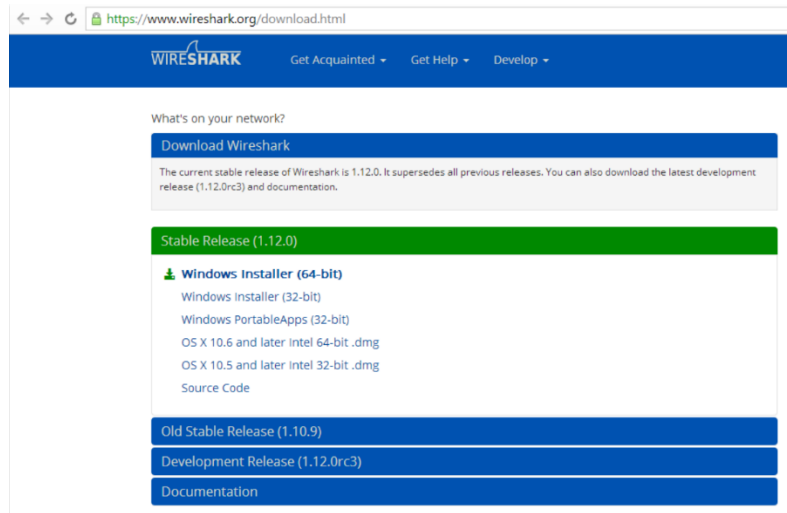
Sau khi download tiến hành cài đặt winpcap



Hình 14: Cài đặt winpcap thành công

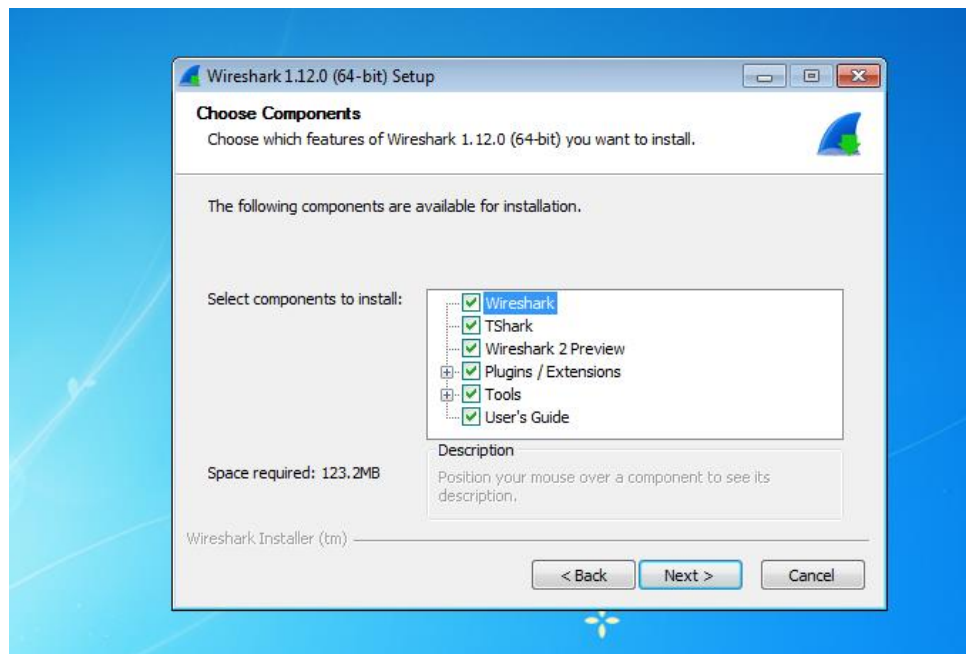
Bước 2: Cài đặt Wireshark

Download và xem thông tin liên quan đến wireshark thông qua trang <https://www.wireshark.org>. Ở trong bài này chúng ta sẽ sử dụng bản wireshark 1.12.0 cho Windows

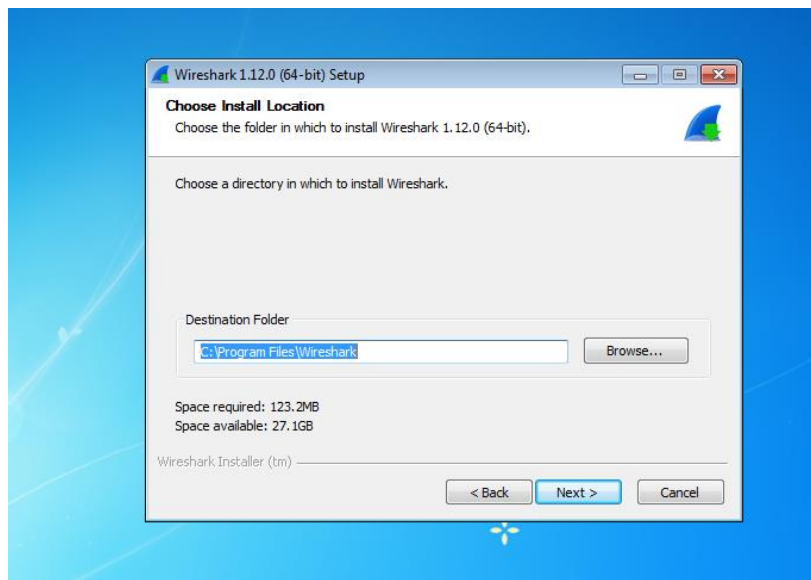


Hình 15: Download wireshark trên trang www.wireshark.org

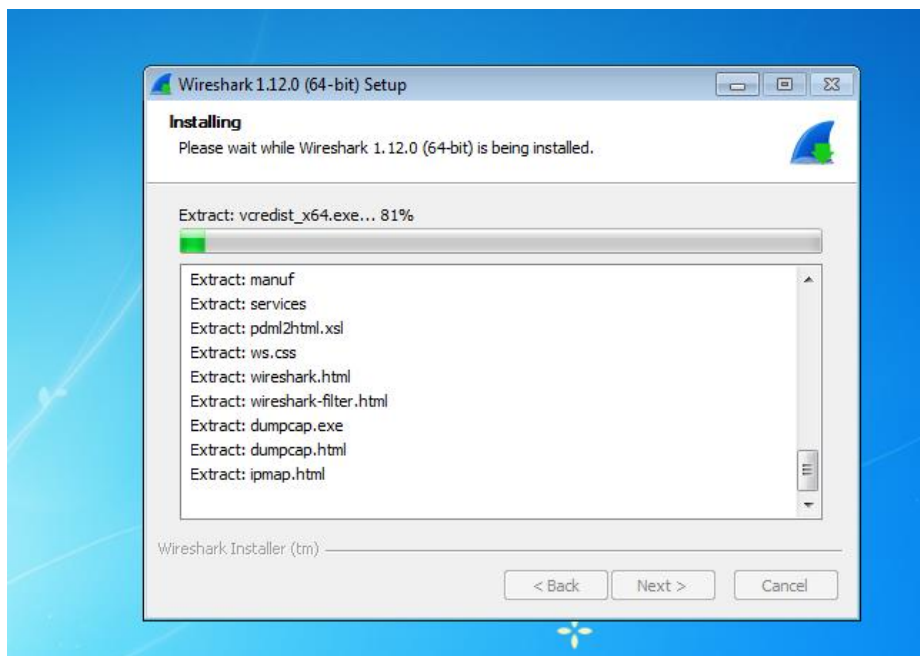
Sau khi download thành công tiến hành cài đặt bằng việc click vào biểu tượng WireShark



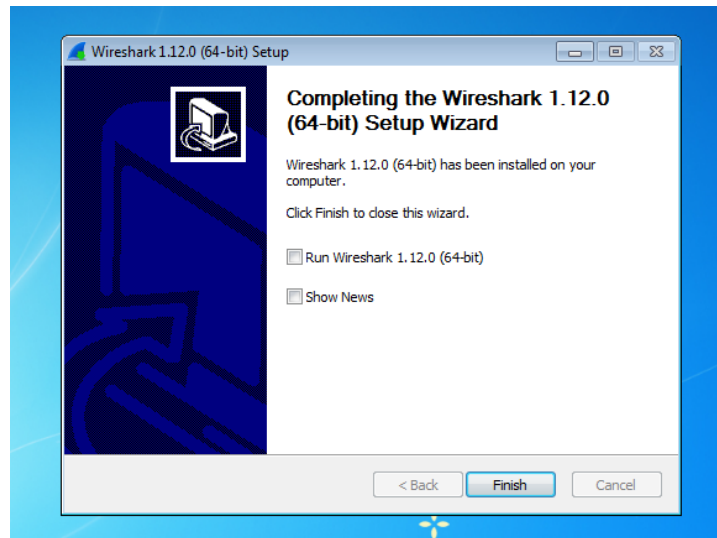
Hình 16: Lựa chọn các component khi cài đặt



Hình 17: Lựa chọn đường dẫn để cài đặt



Hình 18: Install wireshark

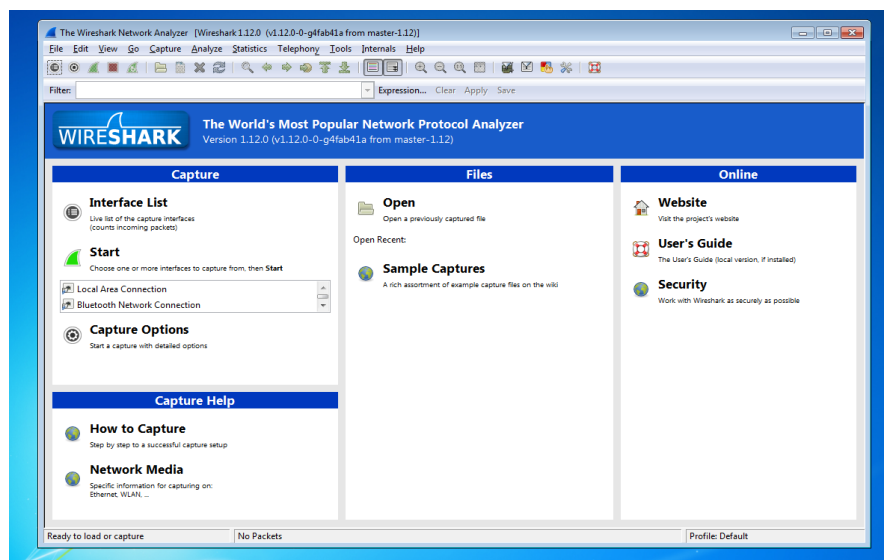


Hình 19: Cài đặt xong wireshark

Bước 3: Sử dụng wireshark

Sau khi cài đặt thành công winpcap và wireshark chúng ta có thể sử dụng nó để chặn bắt gói tin

Wireshark là một công cụ mạnh mẽ hỗ trợ việc chặn bắt gói tin bằng giao diện đồ họa

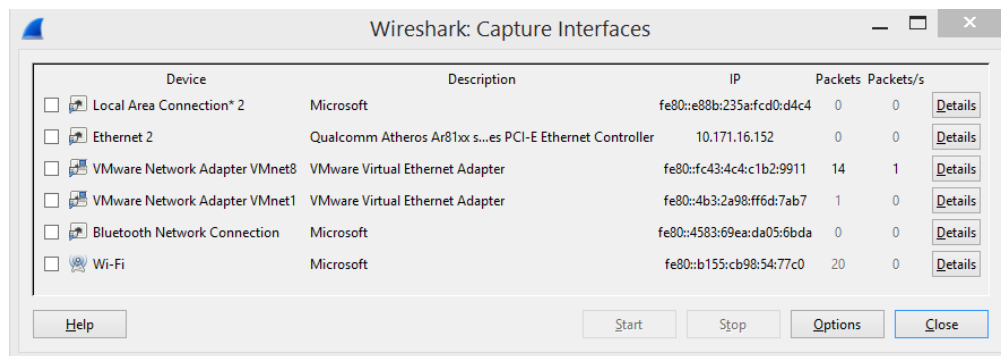


Hình 20: Giao diện chính của wireshark

Ở đây chúng ta sẽ học sử dụng một số chức năng chính của nó

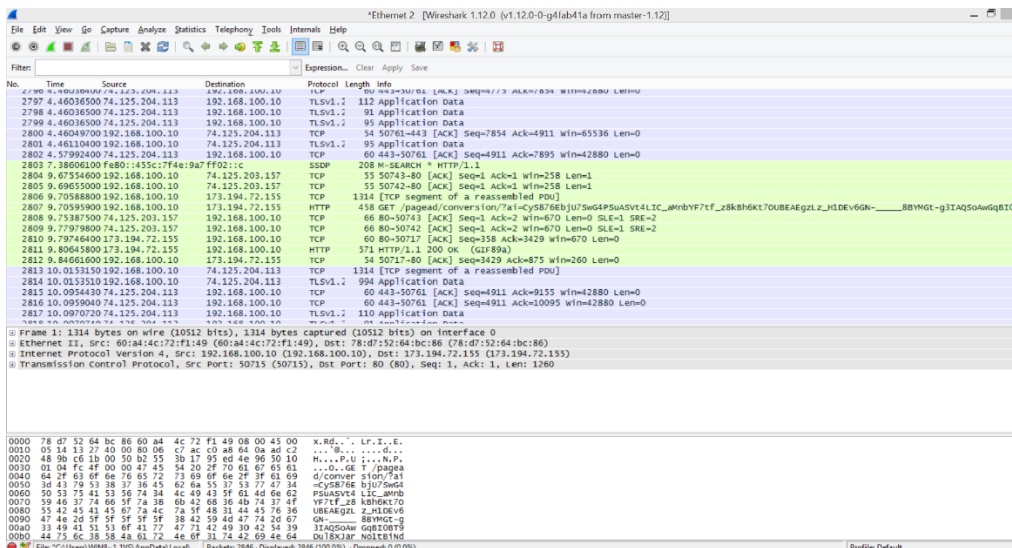
Xem và lựa chọn interface để capture:

- Click vào dòng Interface List ở màn hình chính
- Click vào capture trên thanh công cụ và chọn interface...
- Nhấn tổ hợp phím ctrl + i



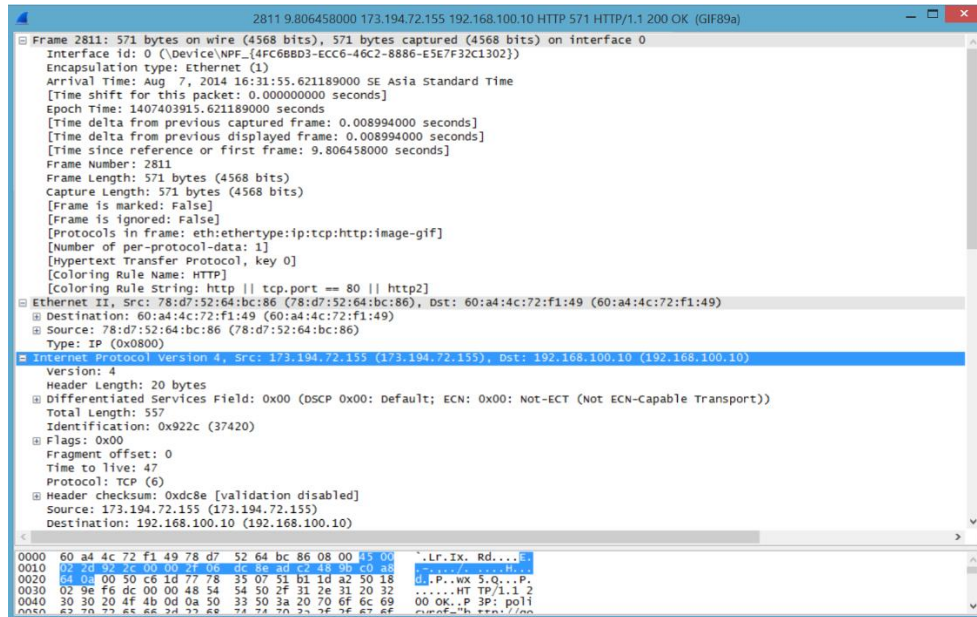
Hình 21: Giao diện hiển thị interface

Sau khi xem interface chúng ta có thể tiến hành chặn bắt gói tin qua interface đó bằng việc chọn interface và nhấn nút Start bắt đầu capture gói tin



Hình 22: Hiển thị các gói tin bắt được trên card ethernet 2

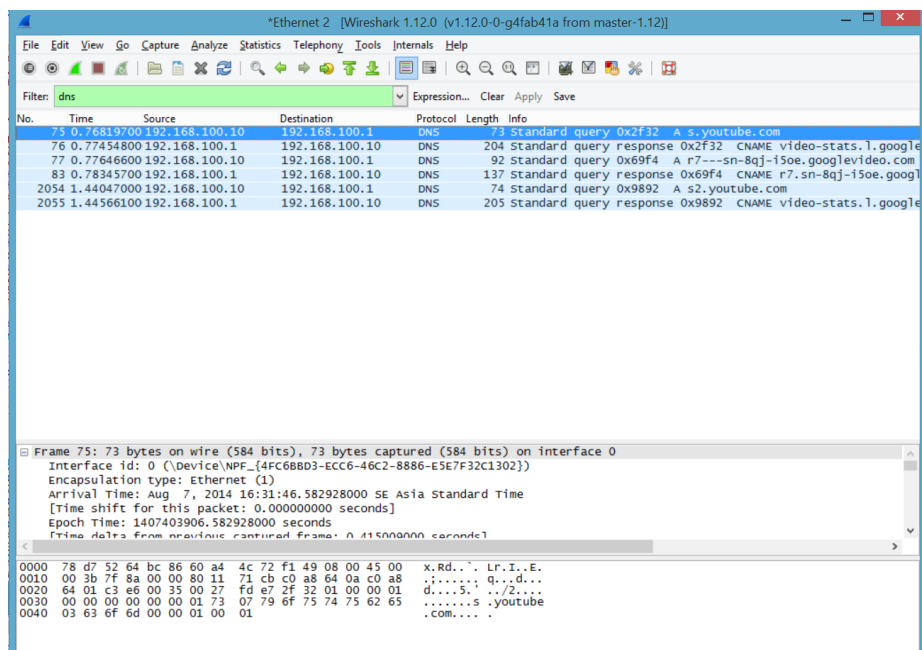
Wireshark hỗ trợ nhiều tính năng nổi bật cho phép bạn xem chi tiết nội dung từng gói tin bằng việc click 2 lần vào gói tin muốn xem nội dung



Hình 23: Xem chi tiết các gói tin

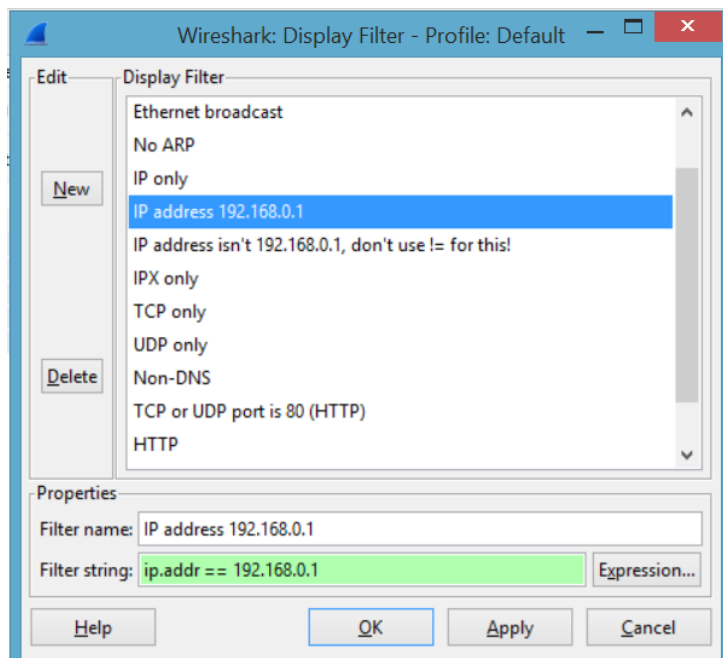
Do số lượng gói tin rất lớn mà không phải gói tin cũng cần thiết hoặc đơn giản bạn chỉ muốn xem một vài gói tin nào đó. Wireshark cung cấp chức năng filter.

Cách cơ bản nhất để áp dụng filter là nhập thông tin vào ô Filter, sau đó nhấn Apply hoặc nhấn Enter. Ví dụ, nếu gõ dns thì chúng ta sẽ chỉ nhìn thấy các gói dữ liệu DNS. Ngay khi nhập từ khóa, Wireshark sẽ tự động hoàn chỉnh chuỗi thông tin này dựa vào gợi ý tương ứng.



Hình 24: filter các gói tin DNS

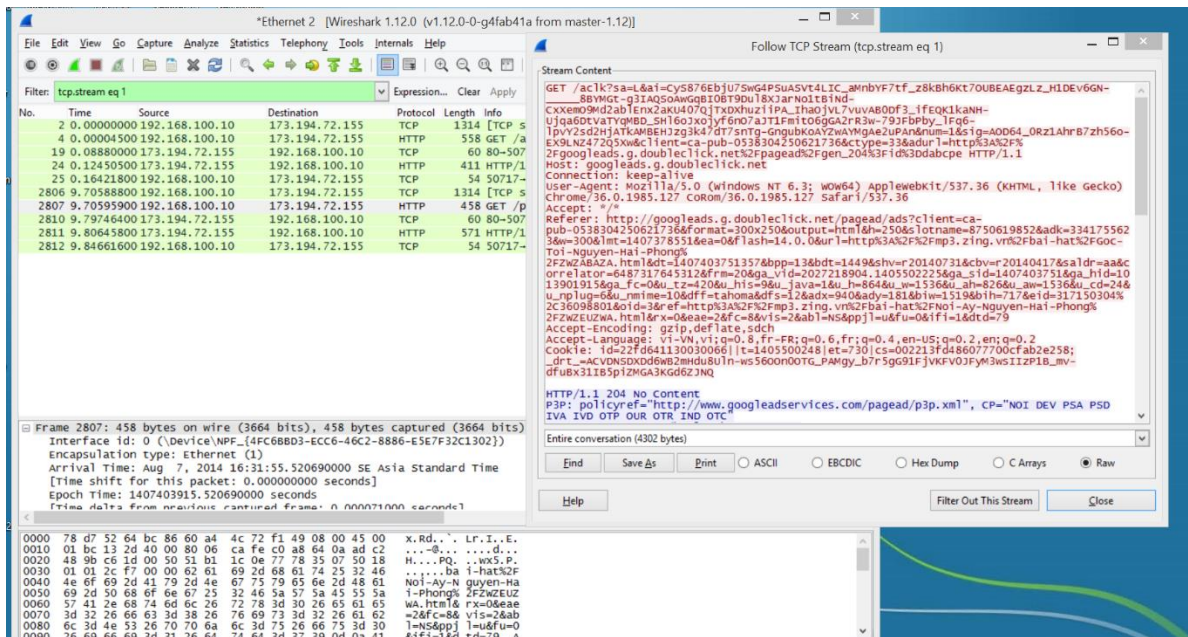
Hoặc nhấn menu Analyze > Display Filters để tạo filter mới:



Hình 25: Tạo filter mới qua giao diện Display Filters

Cách khắc nhân chuột phải vào từng gói tin và chọn **Follow TCP Stream**:

Chúng ta sẽ thấy toàn bộ quãng thời gian giao tiếp giữa server và client, filter sẽ tự động được áp dụng, Wireshark tiếp tục hiển thị đầy đủ và chính xác các gói tin có liên quan:



Hình 26: Follow TCP Stream các gói tin

2.2.3. Ghi nhận phân tích kết quả

❖ Kết quả mong muốn

Cài đặt thành công công cụ chặn bắt gói tin wireshark và thư viện winpcap trên môi trường Windows

Sử dụng được một số chức năng của wireshark

Hiểu được cơ chế làm việc của winpcap và Wireshark

❖ Kết quả thực hiện

Sau khi cài đặt thành công wireshark, tiến hành chạy wireshark bắt các gói tin trên card mạng Ethernet 2

*Ethernet 2 [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]

No.	Time	Source	Destination	Protocol	Length	Info
2797	4.46036500	74.125.204.113	192.168.100.10	TLSv1.2	112	Application Data
2798	4.46036500	74.125.204.113	192.168.100.10	TLSv1.2	91	Application Data
2799	4.46036500	74.125.204.113	192.168.100.10	TLSv1.2	95	Application Data
2800	4.46049700	192.168.100.10	74.125.204.113	TCP	54	50761->443 [ACK] Seq=7854 Ack=4911 Win=65536 Len=0
2801	4.46113400	192.168.100.10	74.125.204.113	TLSv1.2	95	Application Data
2802	4.57992400	74.125.204.113	192.168.100.10	TCP	60	443->50761 [ACK] Seq=4911 Ack=7895 Win=42880 Len=0
2803	7.38066100	192.168.100.10	74.125.204.113	SSDP	208	M-SEARCH * HTTP/1.1
2804	9.67554600	192.168.100.10	74.125.203.157	TCP	55	50741->80 [ACK] Seq=1 Ack=1 Win=258 Len=1
2805	9.69651000	192.168.100.10	74.125.203.157	TCP	55	50742->80 [ACK] Seq=1 Ack=1 Win=258 Len=1
2806	9.70588800	192.168.100.10	173.194.72.155	TCP	1314	[TCP segment of a reassembled PDU]
2807	9.70591900	192.168.100.10	173.194.72.155	HTTP	458	GET /pagead/conversion/7a1-cys876b3j075u6p5u4sv4l7c_mhnyf7tf_z8k8h6kt70UBEAegzr_z_h1Dev6GN-----8BYMGL-g3TAQ50awGqB08
2808	9.75387500	74.125.203.157	192.168.100.10	TCP	66	80->50743 [ACK] Seq=1 Ack=2 Win=670 Len=0 SLE=1 SRE=2
2809	9.77979800	74.125.203.157	192.168.100.10	TCP	66	80->50742 [ACK] Seq=1 Ack=2 Win=670 Len=0 SLE=1 SRE=2
2810	9.78746400	173.194.72.155	192.168.100.10	TCP	60	80->50717 [ACK] Seq=558 Ack=3429 Win=0 Len=0
2811	9.80641800	173.194.72.155	192.168.100.10	HTTP	571	HTTP/1.1 200 OK (GIF89a)
2812	9.84661600	192.168.100.10	173.194.72.155	TCP	54	50717->80 [ACK] Seq=3429 Ack=875 Win=260 Len=0
2813	10.01511500	192.168.100.10	74.125.204.113	TCP	1314	[TCP segment of a reassembled PDU]
2814	10.01511500	192.168.100.10	74.125.204.113	TLSv1.2	994	Application Data
2815	10.09544300	74.125.204.113	192.168.100.10	TCP	60	443->50761 [ACK] Seq=4911 Ack=9155 Win=42880 Len=0
2816	10.09590400	74.125.204.113	192.168.100.10	TCP	60	443->50761 [ACK] Seq=4911 Ack=10095 Win=42880 Len=0
2817	10.09707200	74.125.204.113	192.168.100.10	TLSv1.2	110	Application Data

Frame 11: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits) on interface 0
 Ethernet II, Src: 60:84:4c:72:f1:49 (60:84:4c:72:f1:49), Dst: 78:d7:52:64:bc:86 (78:d7:52:64:bc:86)
 Internet Protocol Version 4, Src: 192.168.100.10 (192.168.100.10), Dst: 173.194.72.155 (173.194.72.155)
 Transmission Control Protocol, Src Port: 50715 (50715), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 1260

0000 78 d7 52 64 bc 86 60 a4 4c 72 f1 49 08 00 45 00 x.Rd...Lr.I..E.
 0010 05 14 17 27 40 08 00 06 c7 ac 0 a8 64 0a ad c2 ..8....q...d...
 0020 48 90 c0 1b 00 50 b2 55 3b 37 95 8b 4e 96 30 10 H...P..l....8...
 0030 01 04 fc 4f 00 00 47 45 54 20 ff 70 01 67 65 61 ...o..ge T /page4
 0040 64 2f 61 6f 6e 76 65 72 73 69 6f 6e 7f 61 69 d/conver sion /a
 0050 3d 41 79 53 38 37 36 45 62 6a 55 37 53 77 67 34 =Cys876E b1u7w6g4
 0060 50 55 75 42 51 56 74 34 4c 49 43 3f 62 4d 6a 62 PSUAQVCS LJC amB
 0070 59 40 37 74 60 3f 7a 38 6b 42 88 36 4b 74 37 4f VF7fF_z8 k8h6kt70
 0080 55 42 45 41 67 74 46 7a 3f 48 31 44 45 76 16 UBEAGdt z_h1Dev6
 0090 47 40 2d 5f 5f 5f 5f 5f 38 42 59 4d 47 74 2d 67 GN-----8BYMGL-g
 00a0 33 40 41 51 5f 6f 41 77 47 71 42 49 30 42 34 39 31AQ50awGqB08
 00b0 44 75 6c 38 58 4a 61 72 4e 6f 31 74 42 69 4e 64 Du18XJAP NOTB1ND

Filter thành công các gói tin DNS

*Ethernet 2 [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]

No.	Time	Source	Destination	Protocol	Length	Info
75	0.76819700	192.168.100.10	192.168.100.1	DNS	73	Standard query 0x2f32 A s.youtube.com
76	0.77454800	192.168.100.1	192.168.100.10	DNS	204	Standard query response 0x2f32 CNAME video-stats.1.google.com
77	0.77646600	192.168.100.10	192.168.100.1	DNS	92	Standard query 0x69f4 A r7---sn-8qj-i5oe.googlevideo.com
83	0.78345700	192.168.100.1	192.168.100.10	DNS	137	Standard query response 0x69f4 CNAME r7.sn-8qj-i5oe.googlevideo.com
2054	1.44047000	192.168.100.10	192.168.100.1	DNS	74	Standard query 0x9892 A s2.youtube.com
2055	1.44566100	192.168.100.1	192.168.100.10	DNS	205	Standard query response 0x9892 CNAME video-stats.1.google.com

Frame 75: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
 Interface id: 0 (\Device\NPF_{4FC6BBD3-ECC6-46C2-8886-E5E7F32C1302})
 Encapsulation type: Ethernet (1)
 Arrival Time: Aug 7, 2014 16:31:46.582928000 SE Asia Standard Time
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1407403906.582928000 seconds
 Time data from previous captured frame: 0.415000000 seconds

0000 78 d7 52 64 bc 86 60 a4 4c 72 f1 49 08 00 45 00 x.Rd...Lr.I..E.
 0010 00 3b 7f 8a 00 00 80 11 71 cb c0 a8 64 0a c0 a8q...d...
 0020 64 01 c3 e6 00 35 00 27 fd e7 2f 32 01 00 00 01 d....s../2....
 0030 00 00 00 00 00 00 01 73 07 79 6f 75 74 75 62 65s.youtube
 0040 03 63 6f 6d 00 01 00 01com....