HUTECH
ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ TP.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC BẢO MẬT THÔNG TIN

TÌM HIỂU GIAO THỨC RADIUS PROTOCOL

Ngành : **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành : **MẠNG MÁY TÍNH**

Giảng viên hướng dẫn : VĂN THIÊN HOÀNG

Nhóm thực hiện: Nhóm 03 Lớp : 10LDTHM1

1. Nguyễn Đăng Hải 1081020022
2. Hoàng Nguyễn Minh Tuấn 1081020115
3. Trần Khánh Mạnh Phương 1081020080
4. Huỳnh Thị Minh Ly 1081020066

TP. Hồ Chí Minh, 2012

MỤC LỤC

-----o)0(o-----

MỞ ĐẦU

Ngày nay, với sự phát triển của xã hội cùng sự tiến bộ vượt bậc về khoa học kỹ thuật, đặc biệt là ngành mạng máy tính, internet đã không còn là khái niệm xa lạ với tất cả mọi người. Hiện nay internet như một món ăn tinh thần không thể thiếu, từ công việc đến học hành, từ thông tin liên lạc đến giải trí, từ tra cứu tìm kiếm thông tin đến giao lưu kết bạn…. Bên cạnh những lợi ích không thể phủ nhận mà internet mang lại vẫn còn đâu đó những mối hiểm họa tiềm ẩn. Một trong những vấn đề đáng lo ngại và được mọi người quan tâm nhất chính là việc bảo mật thông tin trong quá trình sử dụng các dịch vụ, tiện ích của internet. Thông tin, dữ liệu có khả năng bị các kẻ xấu giả mạo hoặc đánh cắp nhằm mục đích phá hoại hay vụ lợi cá nhân. Chẳng hạn có một công ty xây dựng một hệ thống mạng cho phép nhân viên có thể kết nối truy cập tại nhà để sử dụng tài nguyên và dịch vụ. Làm sao để xác định và đảm bảo chỉ có nhân viên của công ty được phép kết nối đến hệ thống. Để giải quyết vấn đề này đòi hỏi cần có một hệ thống đủ thông minh và an toàn để nhận biết đâu là nhân viên của công ty và đâu là những kẻ tấn công từ bên ngoài. Một hệ thống có khả năng xác thực danh tính của người dùng truy cập đến để quyết định người dùng đó được phép kết nối và sử dụng tài nguyên, dịch vụ hay không. Chính xác hơn khi kết nối đến hệ thống người dùng phải xác thực danh tính với một cái máy tính. Chiếc máy tính này sẽ sử dụng một chuỗi các tiến trình và giao thức để xác minh rằng người dùng (máy tính) đang kết nối đến có phải thực sự là nhân viên công ty hay không, cũng như tìm hiểu tất cả những dữ liệu, thông tin nào nhân viên đó được phép truy cập đồng thời có khả năng phản hồi tất cả những thông tin trên để thông báo cho người dùng. Cũng xuất phát từ nhu cầu thiết yếu ấy, giao thức RADIUS (Remote Access Dial In User Service) đã ra đời để đảm nhận những công việc kể trên.

Giao thức Radius hiện nay đang được sử dụng khá rộng rãi, đặc biệt là đối với các nhà cung cấp phân phối mạng ISP và dần dần đã trở thành giao thức tiêu chuẩn cho việc xác thực người dùng truy cập từ xa. Lí do nào mà giao thức Radius lại được sử dụng phổ biến như vậy ? Nhóm chúng em xin chọn đề tài ***"Tìm hiểu về giao thức Radius"*** để tìm hiểu rõ hơn về đặc điểm, nguyên lý hoạt động cũng như các phương thức cũng và thuật toán bảo mật được sử dụng trong giao thức để có câu trả lời cho câu hỏi vừa đặt ra. Đề tài bao gồm 3 chương

Chương 1: Tổng quan về giao thức Radius

Chương 2: Cấu trúc, nguyên lý hoạt động và thuật toán áp dụng

Chương 3: Mô hình thực nghiệm và kết luận

CHƯƠNG 1 : TỔNG QUAN VỀ GIAO THỨC RADIUS

1.1 Những nét chính về giao thức RADIUS:

Radius (Remote Access Dial In User Service) là một giao thức bảo mật mạng dựa theo mô hình Client – Server, hoạt động ở tầng thứ 7 (Application Layer) trong mô hình OSI. Ban đầu giao thức này được phát triển để điều khiển truy cập và xác thực người dùng (thông qua kiểm tra username và password) trong những trường hợp truy cập từ xa (remote access). Tuy nhiên do tính chất là giao thức mở rông nên đã không ngừng cải tiến và mở rộng và ngày càng sử dụng phổ biến rộng rãi cho các máy chủ VPN (Vitual Private Network), các điểm truy cập không dây (wireless network), xác thực chuyển mạch internet, truy cập DSL (Digital Subcriber Line) và các loại truy cập mạng khác. Và là một trong các giao thức khá hữu dụng đối với các nhà quản lý phân phối mạng ISP (Internet Service Provider).

Được thiết kế dựa trên nền tảng kiến trúc AAA (Authentication – Authorization – Accouting) đó là Xác thực - Ủy Quyền – Kế toán và được mô tả khá chi tiết trong tài liệu RFC 2865 (đối với tính năng Authentication – Authorization) và RFC 2866 (với tính năng Accouting). Radius dùng giao thức UDP (User Datagram Protocol) và port 1812 (Authentication- Authorization), 1813 (Accounting) trong quá trình vận chuyển các gói tin. Tuy nhiên trước khi port 1812 và 1813 được chính thức công bố, port 1645 và 1646 đã được khá nhiều Client/Server sử dụng và trở thành cổng mặc định trong suốt thời gian này. Và thói quen đó vẫn còn sử dụng cho đến ngày nay do dó một số server Radius được thiết lập lắng nghe trên cả 2 bộ port này. Radius của Microsoft thiết lập port mặc định là UDP 1812 và UDP 1813, trong khi Cisco thiết lập UDP 1812, 1645 và UDP 1813, 1646 làm port mặc định.

Ngoài ra, Radius còn sử dụng khá nhiều giao thức con hỗ trợ như PPP (Point to Point Protocol), PAP (Password Authentication Protocol), CHAP(Challenge-Handshake Authentication Protocol) , MS-CHAP, MS-CHAP v2(version Microsoft của CHAP)…. cũng như một số thuật toán bảo mật như Share Secret , MD5 ,….

Khi Radius vừa xuất hiện, một câu hỏi lớn được đặt ra đó là vì sao không sử dụng TCP làm giao thức vận chuyển mà thay vào đó là dùng UDP. Nguyên nhân UDP được dùng để phù hợp với những yêu cầu kĩ thuật của Radius:

- Khi một yêu cầu gửi đến một Server Radius thất bại, yêu cầu đó được chuyển cho các Radius server khác. Để thực hiện được yêu cầu này đòi hỏi một bản sao của yêu cầu phải được giữ ở trên tầng giao vận cho phép truyền tải thay thế. Điều này có nghĩa cần thiết bộ định thời gian tái thực hiện yêu câu (retransmission timer)
- Người dùng luôn muốn việc xác thực đăng nhập vào hệ thống diễn ra nhanh chóng. Về mặt này, UDP đáp ứng được về mặt truyền tải dữ liệu nhanh. Đồng thời, giao thức Radius không yêu cầu việc phản hồi khi phát hiện dữ liệu bị mất. Nhờ vậy giảm bớt thời gian khi tiến hành xác thực.
- Bản chất của giao thức Radius là phi trạng thái phù hợp với UDP. Với TCP, client và server phải có mã đặc biệt hoặc giải pháp để giảm thiểu những ảnh hưởng của tổn thất về điện năng, khởi động lại, nghẽn mạng, và ngừng hoạt động của hệ thống. UDP giúp giải quyết điều đau đầu này vì cho phép một phiên làm việc mở trong suốt toàn bộ các giao dịch.

- Để hỗ trợ trong những hệ thống lớn (đôi khi xảy ra trì hoãn yêu cầu , và việc tìm kiếm tốn thời gian) nên Radius yêu cầu hoạt động ở chế độ đa luồng (multithread). UDP có khả năng giúp Server giải quyết nhiều yêu cầu cùng lúc và các khách hàng cũng như thiết bị không bị ảnh huởng lẫn nhau trong quá trình giao dịch.

- Ngoài ra UDP còn giúp giảm lưu lượng đường truyền.

Một điểm yếu của việc sử dụng UDP đó là các nhà phát triển phải tự tạo và quản lý thời gian tái thực hiện lại yêu cầu. So với khá nhiều các ưu điểm vừa nêu trên thì và đây cũng là một hạn chế không lớn lắm nên Radius được quyết định sử dụng UDP.

1.2 **Lịch sử phát triển của giao thức và các RFC liên quan:**

Cũng giống như mọi giao thức khác, Radius được xây dựng từ nhu cầu thiết yếu của con người. Trong trường hợp này, vấn đề đặt ra là cần có một phương pháp có khả năng xác thực, ủy quyền và kế toán cho những người dùng có nhu cầu truy cập tài nguyên máy tính một cách không đồng nhất. Từ năm 1990-1995, hệ thống mạng Merit, một những nhà tiên phong sáng lập nên hệ thống internet, quản lí một số lượng lớn các tài nguyên các thuê bao truy cập quay số (dial-up) trên toàn California. Vào thời điểm đó, phương pháp xác thực được sử dụng khá đặc biệt và cho từng phần cụ thể của thiết bị nên tốn khá nhiều chi phí đầu tư cũng như độ linh hoạt không cao trong việc quản lý và lập báo cáo. Khi số lượng người sử dụng truy cập quay số càng tăng lên, công ty nhận ra cần có một cơ chế linh hoạt hơn và mở rộng hơn so với các đoạn kịch bản (script) cùng các thiết bị độc quyền, khó sử dụng này. Merit đã gửi đi lời yêu cầu đề nghị tìm phương pháp giải quyết vấn đề này, một doanh nghiệp mang tên Livingston đã sớm có lời phúc đáp đầu tiên. Sau đó đại diện của cả hai bên Merit và Livingston đã liên lạc với nhau và kết quả của cuộc hội nghị đó phiên bản đầu tiên của giao thức Radius ra đời. Sau đó đã nhiều phần mềm được xây dựng để hoạt động dựa các thiết bị dịch vụ do Livingston sản xuất và các máy chủ Radius do Merit cung cấp trên nền hệ điều hành Unix. Đến tháng 1/1997 giao thức Radius được xuất bản thành RFC 2058 (Authentication – Authorization) và 2059 (Accounting). Phiên bản tiêu chuẩn hóa hiện nay RFC 2865 và RFC 2866 được phát hành vào tháng 6/2000. Tên của nhà sáng lập ra giao thức Radius, Steve Willins vẫn còn được nhắc tên trong các tài liệu RFC.

Diameter, một giao thức được khởi đầu tại một trong những cuộc họp không chính thức ngay sau khi nhóm làm việc về giao thức Radius được bầu ra, ban đầu giao thức này được dự dịnh như một phiên bản tối ưu (clean version) của Radius. Một số lời kiến nghị gọi đây là Radius v2 nhưng IETF không chấp nhận lời kiến nghị này vì Radius v1 vẫn đang trong thời gian được phê duyệt. Giao thức mới này về sau được gọi là Diameter, hiệu quả gấp đôi so với Radius và được IETF tiêu chuẩn hóa thành bản RFC 3588. Điểm mạnh của Diameter đó là sử dụng giao thức vận tải SCTP và TCP thay cho UDP của Radius.

Tuy nhiên, do đặc tính là một giao thức mở rộng, Radius cũng không ngừng phát triển và mở rộng để trở nên hoàn thiện hơn. Nhiều phần mở rộng đã được thêm vào cùng với các bản RFC bổ sung đã mở rộng cho Radius từ chỉ hỗ trợ truy cập quay số (dial –up) thành hỗ trợ tất cả các dạng xác thực, ủy quyền và kế toán qua mạng. Những phần mở rộng này đã loại bỏ khá nhiều những hạn chế, động lực ban đầu trong

việc tạo ra giao thức Diameter cũng như giảm đi nhiều những tiến bộ trong việc thương mại hóa giao thức này. Ngày nay, Radius đã được sử dụng rộng rãi, đặc biệt là đối với các nhà quản lí cung cấp phân phối mạng internet ISP và dần dần trở thành giao thức tiêu chuẩn đối với người dùng truy cập từ xa.

Tổng hợp một số bản RFC hiện nay của Radius:

- RFC 2865: Authentication & Authorization (Draft Standard)
- RFC 2866: Accounting (Information)
- RFC 2867: Accounting extensions for Tunneling (Information)
- RFC 2868: Attributes for Tunneling (Information)
- RFC 2869: RADIUS Extensions (Information)
- RFC 3162: RADIUS and IPv6 (Proposed Standard)
- RFC 3575: IANA Considerations (Proposed Standard)
- RFC 3576: Dynamic Authorization Extensions (Information)
- RFC 3579: RADIUS Support for EAP (Information)
- RFC 3580: IEEE 802.1X Usage Guidelines (Information)

1.3 Kiến trúc AAA:

Một kiến trúc AAA đơn thuần một thiết kế được vạch ra để đảm bảo cho các thành phần có thể hoạt động phù hợp với nhau. Việc triển khai một mô hình kiến trúc AAA phức tạp hay đơn giản tùy vào môi trường sử dụng. Chính xác hơn kiến trúc này được thiết kế để làm việc trong những môi trường đa dạng về yêu cầu của người dùng cũng như đa dạng về các thiết kế hệ thống mạng. Để đạt được khả năng này thì kiến trúc AAA cần phải có một số thuộc tính quan trọng.

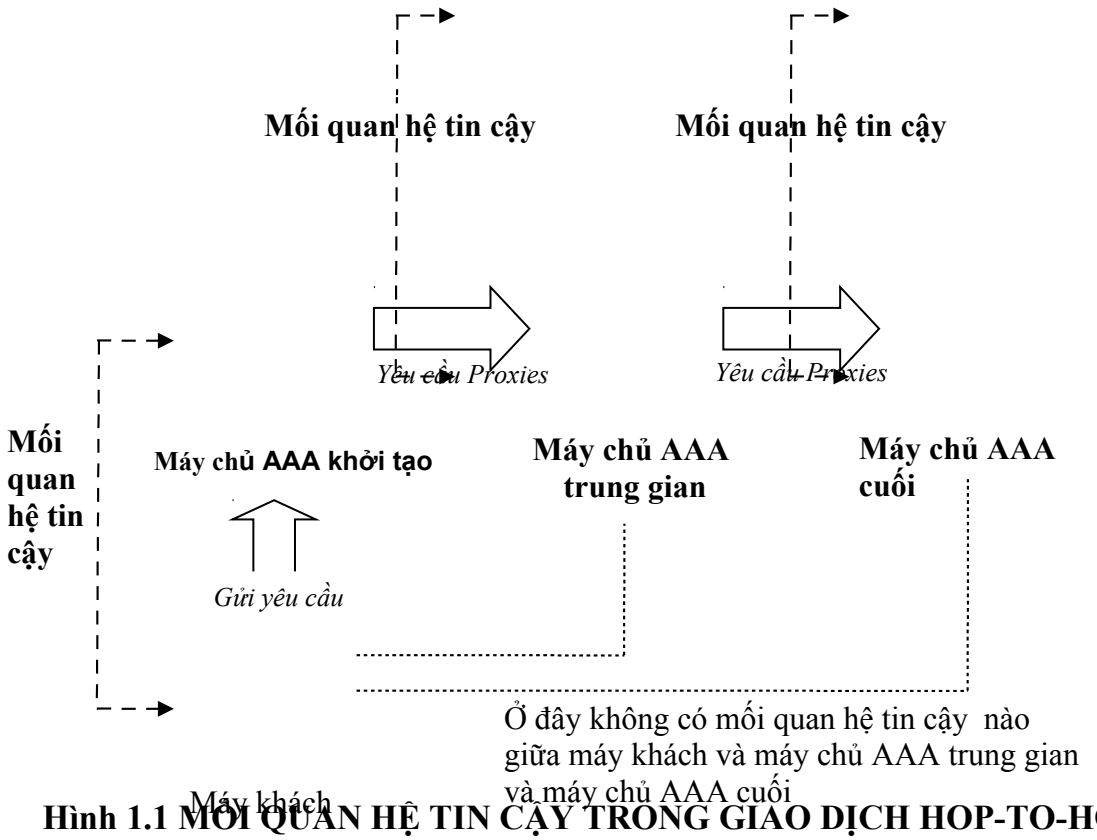
Đầu tiên, mô hình AAA phụ thuộc vào sự tương tác của máy khách và máy chủ, trong trường hợp này là một máy khách gửi yêu cầu về việc sử dụng một tài nguyên hay một dịch vụ nào đó trên một hệ thống máy chủ. Môi trường Client- Server đảm bảo cho một thiết kế cân bằng tải tốt, đáp ứng được 2 yếu tố quan trọng đó là khả năng sẵn sàng phục vụ và thời gian phản hồi. Đặc biệt là máy chủ có thể được phân cấp và phân phối trong toàn hệ thống mạng.

Một yếu tố cũng ảnh hưởng đến kiến trúc AAA này đó là proxy. Một máy chủ AAA được cấu hình để cấp quyền cho một yêu cầu gửi đến hoặc cho phép yêu cầu đó đi qua để đến một máy chủ AAA khác, quá trình lập lại đến khi nào đến đúng được máy chủ thích hợp để cấp quyền về tài nguyên cho yêu cầu đó. Một chuỗi proxy được tạo ra và trong đó các máy chủ AAA gửi đi yêu cầu từ client cũng như từ các máy chủ trung gian khác. Chính vì thế mô hình đòi hỏi phải có mối quan hệ tin tưởng giữa client / server cũng như giữa các server AAA với nhau.

Khi một client gửi yêu cầu cần truy cập đến một dịch vụ và tài nguyên nào đó từ một máy chủ AAA (client ở đây bao hàm cả các AAA proxy trung gian), để đến được máy chủ AAA cần thiết phải thông qua 1 trong 2 hình thức giao dịch đó là hop-to-hop và end-to-end.

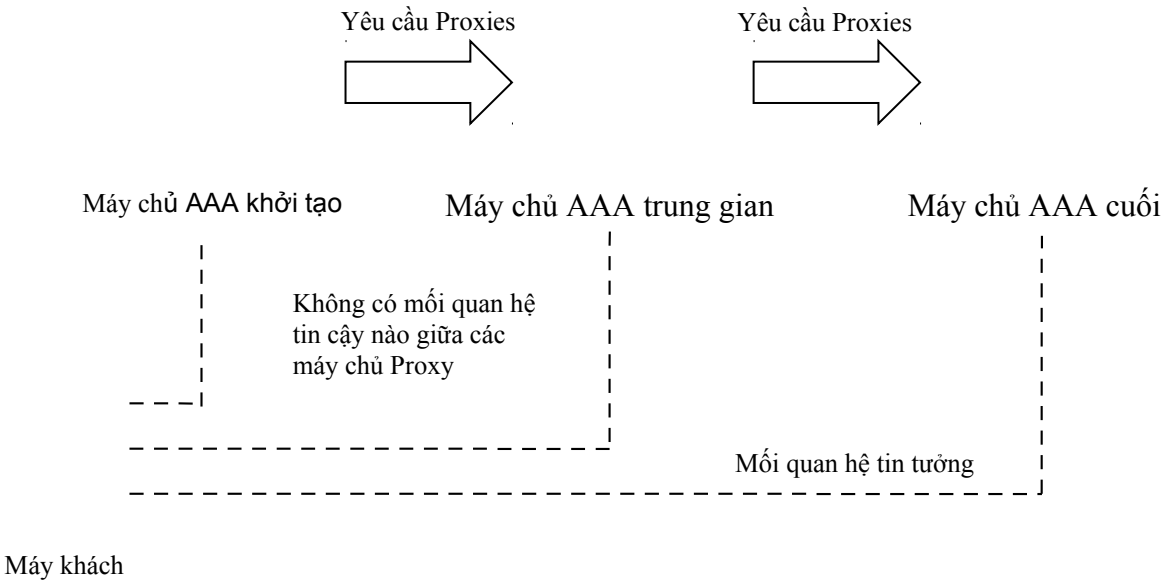
Đối với giao dịch hop-to-hop, khi client gửi một yêu cầu khởi tạo đến một thiết bị AAA. Một mối quan hệ tin cậy sẽ được thiết lập giữa client và server AAA đó. Và server xác định rằng yêu cầu này cần phải được chuyển đến một server AAA ở khu vực khác vì vậy sẽ hoạt động như một proxy và tiến hành liên lạc. Tiếp đến sẽ phải

thiết lập mối quan hệ tin cậy giữa 2 server AAA này, server AAA hoạt động như một proxy lúc này được xem như một client và server AAA cần liên lạc đến đóng vai trò của một server. Tuy nhiên mối quan hệ tin cậy không được phép kế thừa, tức là giữa client khởi tạo yêu cầu và server AAA thứ 2 không có mối quan hệ tin cậy.



Hình 1.1 MỐI QUAN HỆ TIN CẬY TRONG GIAO DỊCH HOP-TO-HOP

Sự khác biệt giữa 2 hình thức giao dịch hop-to-hop và end-to-end là là cách thiết lập mối quan hệ tin cậy trong mô hình. Giữa các server lúc này không cần phải thiết lập mối quan hệ tin cậy với nhau nữa, mà chủ yếu tập trung vào mối quan hệ tin cậy giữa khách hàng tạo yêu cầu và máy chủ AAA cuối cùng. Và do không có mối quan hệ tin cậy giữa các server nên việc gửi các thông tin nhạy cảm thông qua các yêu cầu proxy giữa các server đòi hỏi phải có sự xác thực và đảm bảo về tính toàn vẹn dữ liệu. Thông thường để giải quyết vấn đề này, người ta sử dụng giấy chứng nhận số hoặc các chứng nhận của hệ mã khác công khai (PKI – Public Key Infrastructure).



Hình 1.2 MỐI QUAN HỆ TIN CẬY TRONG GIAO DỊCH END-TO-END

1.3.1 Xác thực (Authentication):

Xác thực là quá trình xác minh danh tính của một người (hay một máy tính). Hình thức xác thực phổ biến thường thấy nhất hiện nay là sự kết hợp giữa tên đăng nhập (ID logon) và mật khẩu (password). Trong đó phần mật khẩu sẽ đại diện cho tính xác minh về người sử dụng. Tuy nhiên, việc phân phối mật khẩu sẽ phá hủy phương pháp xác thực này. Do đó những trang web thương mại điện tử và các giao dịch kinh doanh thông qua internet đòi hỏi phải có một cách thức xác thực phù hợp, mạnh mẽ và đáng tin cậy hơn. Và giấy chứng nhận số (digital certificate) là một trong những hướng giải quyết phù hợp, giấy chứng nhận số là một cái gì đó chứng nhận bạn là ai trên internet hoặc trong mạng nội bộ. Giấy chứng nhận số có thể là một số ID cá nhân (personal ID) hoặc là một chữ kí điện tử (digital signature) được dùng để chứng minh tính xác thực hoặc đảm bảo một thông báo được gửi đúng từ người mà bạn cho rằng đã xuất phát từ đó và không bị thay đổi trong suốt quá trình truyền đi.

Một giấy chứng nhận số xác minh tính xác thực của người đang tải thông tin đến một máy chủ an toàn. Ngoài ra, một server cũng cần giấy chứng nhận để xác minh tính xác thực đối với người truy cập, nghĩa là cần phải sự xác minh từ 2 phía để đảm bảo về tính an toàn. Giấy chứng nhận được phát hành bởi CA (certificate authority – người có thẩm quyền cấp giấy chứng nhận), là một tổ chức đáng tin cậy nhằm xác nhận các chứng từ và đóng dấu xác nhận lên những chứng từ này. Toàn bộ quá trình xác nhận sử dụng một cơ chế an toàn, đã biết và đáng tin cậy, dựa trên hệ mã hóa dùng khóa công khai. Một trong những công ty chứng nhận có uy tín nhất trên thế giới hiện nay đó là Verisign.

Định nghĩa đơn giản nhất về giấy chứng nhận số là một vật chứa (container) gồm khóa công khai của người dùng và đôi khi là một vài thông tin của người dùng đó. Và vật chứa này sau đó được ký bởi một CA đáng tin cậy dùng các khóa riêng của

nó. Về cơ bản, CA chấp nhận giấy chứng nhận và mã hoá nội dung. Sau đó các giấy chứng nhận này được dùng cho các giao dịch kinh doanh cũng như để đăng nhập an toàn vào máy chủ. Hệ thống chấp nhận giấy chứng nhận có thể nhận được một bản sao các khoá công khai của CA và kiểm tra rằng giấy chứng nhận là đúng.

Tóm lại mục đích chính của việc xác thực đó là tạo một mối quan hệ đáng tin cậy giữa hai đối tượng tham gia, hay chính xác hơn là giữa hai người dùng hợp lệ. Việc tạo ra mối quan hệ tin cậy giữa 2 hệ thống này cho phép thực hiện các chức năng quan trọng đó máy chủ proxy, tại máy chủ này hệ thống chấp nhận các yêu cầu từ một đại diện của hệ thống khác và cho phép triển khai thực hiện cơ chế AAA để mở rộng thành hệ thống không đồng nhất để hỗ trợ cho nhiều loại khách hàng và dịch vụ khác nhau.

1.3.2 Ủy quyền (Authorization):

Một bước khá quan trọng tiếp theo của việc xác thực đó là Ủy quyền. Việc ủy quyền này có liên quan đến sử dụng một số các tập hợp luật lệ (rule) hoặc một số kiểu mẫu (templates) để quyết định xem những người dùng đã qua được khâu xác thực được phép làm gì trên hệ thống. Ví dụ như, đối với trường hợp một nhà quản lý phân phối mạng internet ISP, việc uỷ quyền sẽ quyết định xem thuê bao tài khoản nào được cấp phát IP tĩnh thay vì phải do hệ thống DHCP cấp phát như thường lệ. Và những luật lệ này do chính nhà quản trị hệ thống (administrator) định nghĩa.

Một hệ thống máy chủ AAA đòi hỏi cần phải có sự logic và nhạy bén trong việc giải quyết các yêu cầu. Nghĩa là khi một yêu cầu gửi đến, hệ thống cần phải phân tích xem đâu là những yêu cầu hợp lệ và đồng thời cấp quyền truy cập đến bất cứ những gì được phép truy cập. Chẳng hạn, một thuê bao khách hàng truy cập quay số yêu cầu một đa kết nối (multilink) đến server. Một hệ thống AAA bình thường sẽ đơn giản từ chối yêu cầu này nhưng một hệ thống tốt và đủ thông minh sẽ tiến hành phân tích yêu cầu, và xác định rằng một khách hàng chỉ được phép có một kết nối quay số đến, sau đó sẽ mở một kênh để kết nối với khác hàng này trong khi từ chối các kênh khác.

1.3.3 Kế Toán (Accounting) :

Tính năng quan trọng còn lại của kiến trúc AAA đó chính là kế toán, đây là tính năng dùng để tính toán và tài liệu hoá lại về tất cả những tài nguyên mà một người dùng đã sử dụng trong suốt quá trình truy cập. Điều này bao cả việc tính toán được lưu lượng thời gian hệ thống cũng như là dung luợng của người dùng trong quá trình gửi và nhận trong một phiên làm việc. Thực ra kế toán là quá trình ghi nhận lại số liệu thống của các phiên làm việc cũng như việc sử dụng thông tin để phục vụ cho việc kiểm soát uỷ quyền truy cập, thanh toán, phân tích các khuynh hướng sử dụng, sử dụng tài nguyên hiệu quả …..

Các dữ liệu của việc kế toán có khá nhiều mục đích sử dụng. Một nhà quản trị có thể phân tích các yêu cầu thành công để xác định và dự đoán về năng lực tải dữ liệu của hệ thống trong tương lai. Một người chủ kinh doanh có thể theo dõi các dịch vụ đang được sử dụng và thông qua đó sẽ điều chỉnh để có mức thanh toán hợp lý. Một chuyên gia an ninh có thể xem xét về các yêu cầu bị từ chối, kiểm tra xem có những gì khả nghi hoặc không bình thuường hay không, nhờ đó có thể sớm chặn được những tay

tin tặc hay nhưng kẻ tải lậu dữ liệu (freeloader). Tóm lại các dữ liệu kế toán này khá hữu ích đối với một quản trị viên của hệ thống máy chủ kiến trúc AAA

1.4 **Dịch vụ an ninh Radius hỗ trợ:**

Được xây dựng dựa trên nền tảng kiến trúc AAA nên Radius thừa hưởng tất cả các tính chất cũng như các chức năng bao gồm cả xác thực, ủy quyền và kế toán. Tuy nhiên với tính chất là một giao thức bảo mật mạng, các chức năng này phải được xây dựng để phục vụ các dịch vụ an ninh. Hiện nay hầu hết tất giao thức bảo mật mạng được xây dựng theo chuẩn chung đó là X.800 hoặc RFC 2828. Theo chuẩn X.800 có 5 hình thức bảo mật bao gồm: xác thực, điều khiển truy cập, toàn vẹn dữ liệu, bảo mật thông tin và chống chối bỏ. Điểm mạnh của Radius đó là hỗ trợ hoàn toàn 5 hình thức bảo mật trên

- ***Xác thực:*** xác minh danh tính của một người (hay một máy tính) tham gia truy cập vào hệ thống để thiết lập mối quan hệ tin cậy (trust relationship). Để phục vụ dịch vụ này, Radius cho phép hỗ trợ sử dụng khá nhiều giao thức xác thực như PAP, CHAP, MS-CHAPv1,MS-CHAPv2, EAP v.v….
- ***Điều khiển truy cập:*** kết hợp với bước xác thực ở trên hệ thống tiến hành phân tích và xác định các yêu cầu gửi đến để cấp quyền truy cập đến những tài nguyên cũng như dịch vụ đối với những yêu cầu hợp lệ. Đồng thời ngăn chặn những yêu cầu truy cập trái phép từ bên ngoài.
- ***Toàn vẹn dữ liệu:*** Radius sử dụng cơ chế bí mật chia sẻ (share secret) để đảm bảo dữ liệu không bị thay đổi trong quá trình vận chuyển.
- ***Bảo mật dữ liệu:*** Radius sử dụng khóa *secret* (share secret) kết hợp với hàm băm MD5 và trường *Authenticator Request* để mã hóa bảo vệ cho trường thông tin *user-password* trong quá trình truyền tải. Ngoài ra toàn bộ quá trình truyền tải đều tiến hành trong mạng nội bộ nên giảm thiểu nguy cơ tấn công.
- ***Chống chối bỏ:*** Radius sử dụng trường thông tin *Authenticator Response* trong các gói tin để đảm bảo các gói tin gửi đi là của server. Riêng các cầu yêu cầu Access-Request đòi hỏi phải cấu hình sử dụng trường *Message-Authenticator* để thực hiện dịch vụ này.

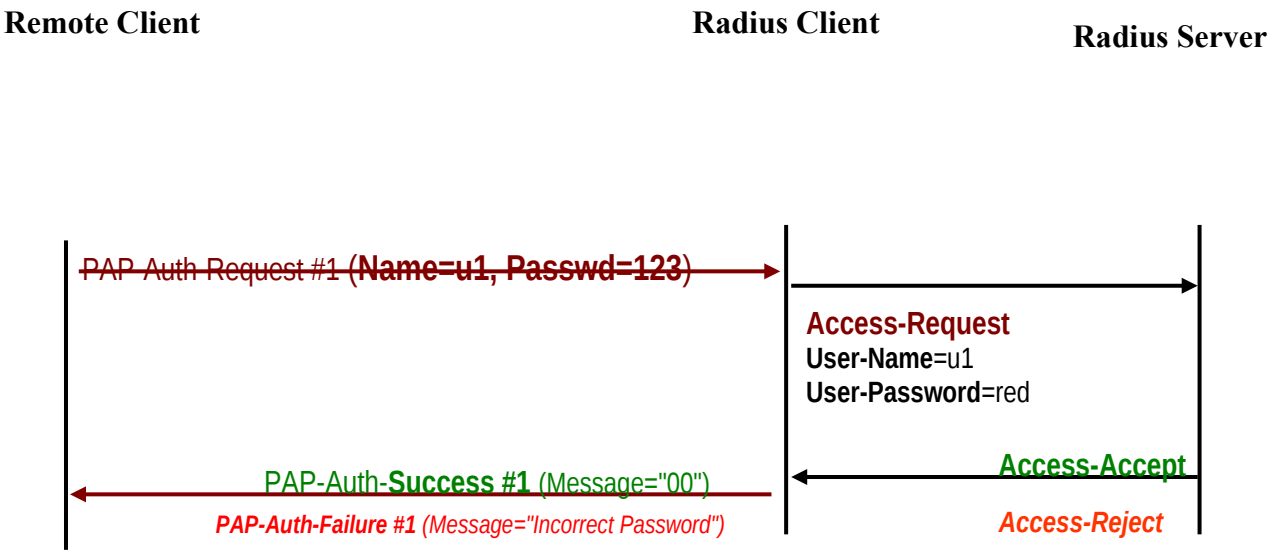
1.5 Một số phương thức và thuật toán trong Radius

1.5.1 Giao thức xác thực :

• *PAP (Password Authentication Protocol):* là một cơ chế xác thực người dùng thông qua việc sử dụng trường thông tin password. PAP sử dụng giao thức PPP (Point to Point Protocol) để xác nhận người dùng trước khi cho phép truy cập vào tài nguyên của hệ thống. Hầu hết các hệ thống xác thực cho người dùng từ xa đều hỗ trợ cơ chế này. Cơ chế của PAP được xem là không an toàn vì quá trình truyền tải mã ASCII của password qua mạng không được mã hóa. Vì vậy nếu ko hỗ trợ các giao thức xác thực khác như CHAP hoặc EAP thì mới dùng đến giao thức này.

Giao thức xác thực dựa trên password là giao thức mà 2 bên tham gia chia sẻ một password từ trước và dựa vào password đó làm cơ sở để xác thực. Hiện tại hệ thống xác thực dựa trên mật khẩu được phân thành 2 loại : hệ thống xác thực mật khẩu yếu và mạnh (weak-password authentication và strong-password authen tication).

Trong đó giao thức xác thực mật khẩu mạnh có lợi thế hơn các so với xác thực mật khẩu yếu, trong đó chi phí tính toán của họ là nhẹ hơn, thiết kế đơn giản, và thực hiện được dễ dàng hơn, và do đó đặc biệt thích hợp đối với một số môi trường hạn chế.

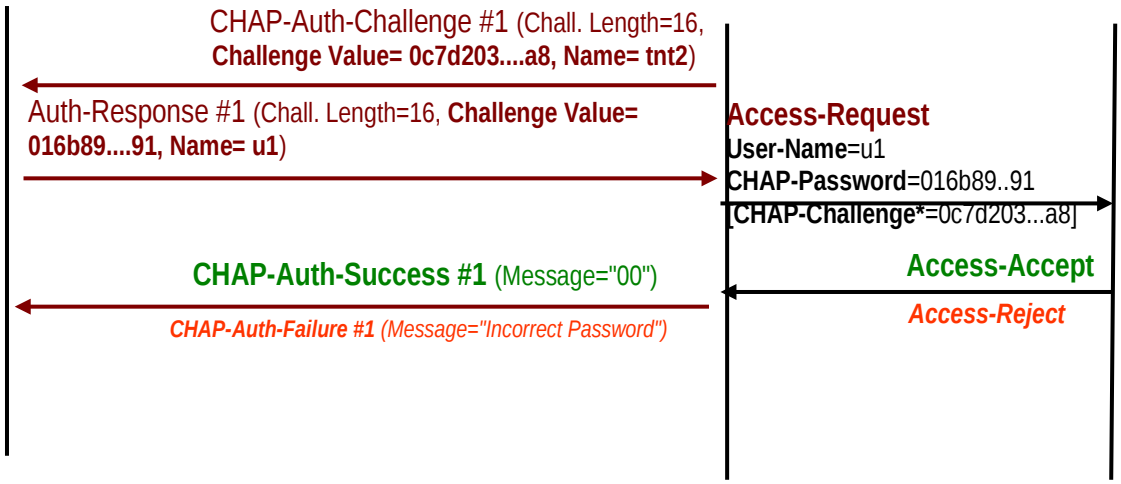


HINH 1.3 MÔ HÌNH MINH HỌA RADIUS DÙNG CƠ CHẾ XÁC THỰC PAP

Cơ chế làm việc:

- User sẽ gửi lên hệ thống username và password
- Server sẽ phản hồi authentication-ack (nếu thông tin phù hợp) hoặc authentication–nak nếu không phù hợp

• *CHAP (Challenge /Handshake Authentication Protocol)* : cơ chế cho phép ngăn chặn tấn công theo kiểu replay attack (Replay attack là dạng thức tấn công mà hacker chặn đứng dòng dữ liệu thay đổi nó và truyền lại cho người nhận mà người nhận vẫn không biết gì về điều này) bằng cách thông qua bộ đếm các bước thay đổi (incrementally changing identifier) và một biến giá trị challenge(Challenge value). CHAP yêu cầu cả 2 bên tham gia phải nắm được bản rõ của của một giá trị bí mật mặc dù giá trị này không bao giờ được gửi qua mạng. Đối với các phiên bản MS-CHAP thì không yêu cầu cả 2 bên biết bản rõ, tuy nhiên lại vấp phải một số nhược điểm khác.Cũng như PAP, CHAP dùng giao thức PPP để xác nhận danh tính của người dùng truy cập từ xa . CHAP sẽ định kì xác minh danh tính của người dùng bằng cách sử dụng cơ chế bắt tay ba bước. Điều này xảy ra tại thời điểm thiết lập liên kết ban đầu (LCP- Link Control Protocol), và có thể xảy ra một lần nữa bất cứ lúc nào sau đó. Xác minh được dựa trên một bí mật được chia sẻ (share secret) (chẳng hạn như mật khẩu của người dùng).

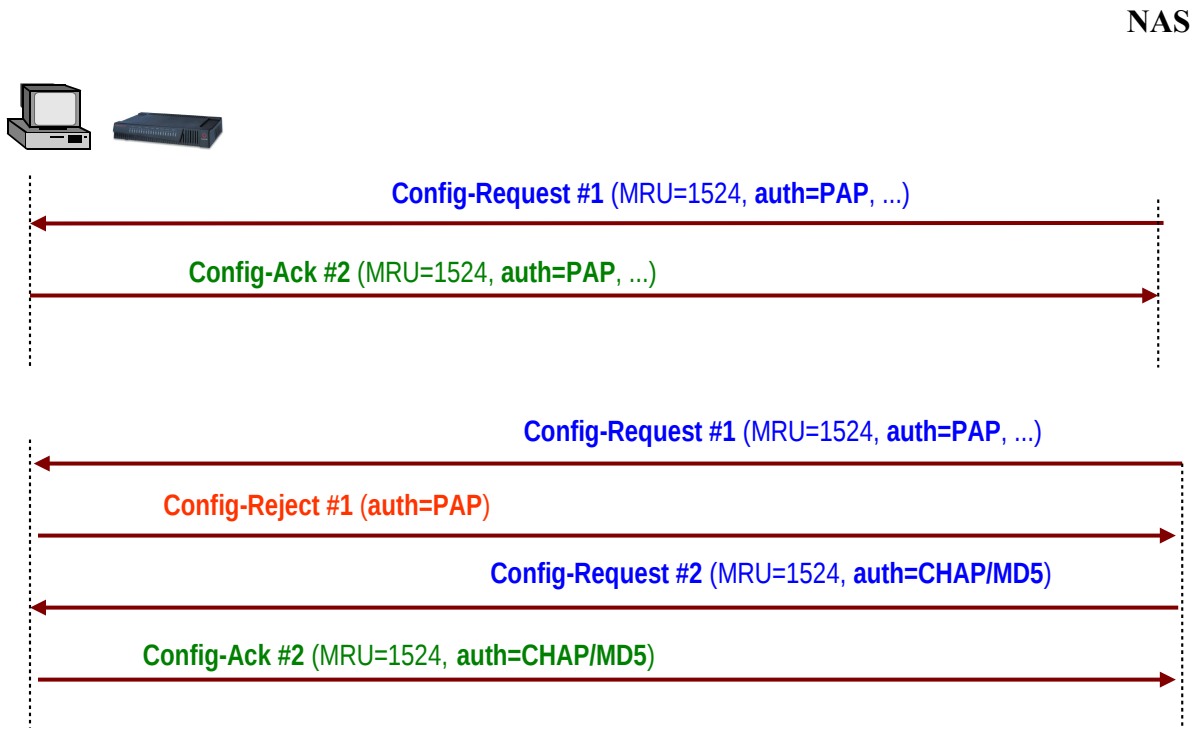


HÌNH 1.4 MÔ HÌNH MINH HỌA RADIUS DÙNG CƠ CHẾ XÁC THỰC CHAP

Cơ chế làm việc:
- Sau khi liên kết được thiết lập hệ thống sẽ gửi một thông điệp challenge đến người dùng.
- Người dùng sẽ phản hồi bằng một giá trị được tính bằng cách sử dụng một hàm băm một chiều trên giá trị challenge nhận được và kết hợp với mã bí mật chia sẻ.
- Tại hệ thống cũng tiến hành tự tính toán giá trị tương tự như của người dùng. Nếu các giá trị phù hợp, thực hiện xác nhận chứng thực, nếu không sẽ chấm dứt kết nối.
- Sau một khoảng thời gian ngẫu nhiên hệ thống lại tiến hành thực hiện chứng thực gửi một challenge mới đến người dùng và các bước được lặp đi lặp như vậy.

• MS-CHAP v1,v2 (các phiên bản CHAP của Microsoft):

Những điểm khác nhau giữa phiên bản MS-CHAP và CHAP :
- Được thết kể phù hợp với sản phẩm của Windows
- Không đòi hỏi lưu trữ bản rõ hay bản đã mã hoá của password
- Cung cấp cơ chế xác thực lặp lại và cơ chế chuyển đổi password
- Định nghĩa 1 dãy code phục vụ cho việc phản hồi khi gửi thông điệp thất bại



HÌNH 1.5 MÔ HÌNH MINH HỌA
BẮT TAY LCP XÁC ĐỊNH PHƯƠNG THỨC XÁC THỰC

• EAP(Extensible Authentication Protocol) thường được sử dụng để xác thực trong các hệ thống mạng không dây với kết nối PPP. Điểm chính của phương thức xác thực này là dùng khoá và các tham số tùy theo các phương thức mà EAP hỗ trợ (EAP-MD5, EAP-OTP, EAP-GTC, EAP-TLS, EAP-IKEv2, EAP-SIM, EAP-AKA and EAP-AKA ….). Mỗi giao thức sử dụng định nghĩa một cách để đóng gói các thông điệp EAP trong tin nhắn của giao thức.

1.5.2 Cơ chế share secret :

Một bí mật chia sẻ (share secret) là một đoạn chuỗi được xem như một mật khẩu giữa :

- Radius Client và Radius Server
- Radius Client và Proxy Radius
- Proxy Radius và Radius Server

Trong trường hợp hệ thống bao gồm một Radius Client, một Proxy Radius, và một Radius Server, thì mã chia sẻ bí mật được sử dụng giữa Radius client và Proxy Radius có thể là khác so với mã bí mật chia sẻ được sử dụng giữa Proxy Radius và Radius Server.

Share secret được dùng để xác minh các thông điệp Radius (ngoại trừ Access Request) trong quá trình trao đổi giữa các máy đã thực hiện chia sẻ mã bí mật từ trước. Bí mật được chia sẻ cũng xác nhận rằng các thông điệp Radius này đã không bị sửa đổi trong quá trình vận chuyển (đảm bảo tính toàn vẹn dữ liệu). Ngoài ra mã chia sẻ bí mật cũng được sử dụng để mã hóa một số thuộc tính trong thông điệp Radius, chẳng

hạn User-Password và Tunel-Password. Để thực hiện tính xác minh cho thông điệp Access-Request, phải cấu hình sử dụng trường thông tin Messeage -Authenticator trên NAS (Radius Client) và cả Remote user.

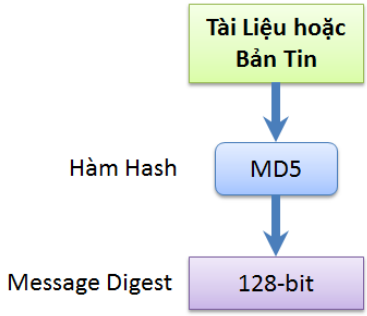
Những điểm lưu ý khi tạo và sử dụng mã chia sẻ bí mật:

- Mã chia sẻ bí mật là thông tin nhạy cảm được chia sẽ dùng chung giữa 2 máy Radius (Client và Server)
- Nếu trong hệ thống cấu hình proxy Radius, giữa các cặp Radius Client/server phải sử dụng mã chia sẻ bí mật khác nhau.
- Nên sử dụng mã chia sẻ bí mật có độ dài hơn 22 kí tự để đảm bảo an toàn
- Để tạo một mã chia sẻ bí mật an toàn nên tạo 1 chuỗi ngẫu nhiên bao gồm kí tự hoa và thường (a-z, A-Z), kí tự số (0-9) và kí tự hình (!, ? , : ,….) và thường xuyên thay đổi mã chia sẻ bí mật này.

Ví dụ về một mã chia sẻ bí mật an toàn: 8d#>9fq4bV)H7%a3-zE13sW

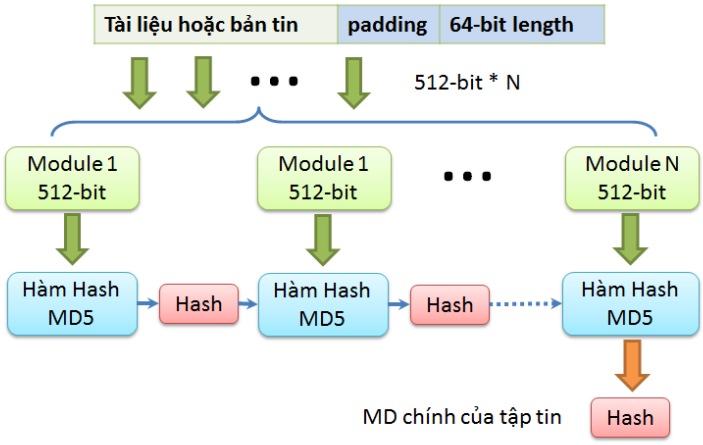
1.4.2.3 Hàm băm MD5 :

MD5 (**Message-Digest algorithm 5**) là một Bộ tạo Hash mật mã được sử dụng phổ biến với giá trị Hash dài 128-bit. Là một chuẩn Internet (RFC 1321), MD5 đã được dùng trong nhiều ứng dụng bảo mật, và cũng được dùng phổ biến để kiểm tra tính toàn vẹn của tập tin. Một bảng băm MD5 thường được diễn tả bằng một số hệ thập lục phân 32 ký tự.



*** ĐẶC ĐIỂM MD5:**

- Việc tính MD đơn giản, có khả năng xác định được file có kích thước nhiều Gb.
- Không có khả năng tính ngược, khi tìm ra MD.
- Do bản chất ngẫu nhiên của hàm băm và số lượng cực lớn các giá trị hash có thể, nên hầu như không có khả năng hai bản tin phân biệt có cùng giá trị hash.
- Giá trị MD phụ thuộc vào bản tin tương ứng.
- Một chuỗi chỉ có duy nhất một hash.
- Giá trị MD phụ thuộc vào tất cả các bit của bản tin tương ứng.



HÌNH 1.6 MD5

Ví dụ :

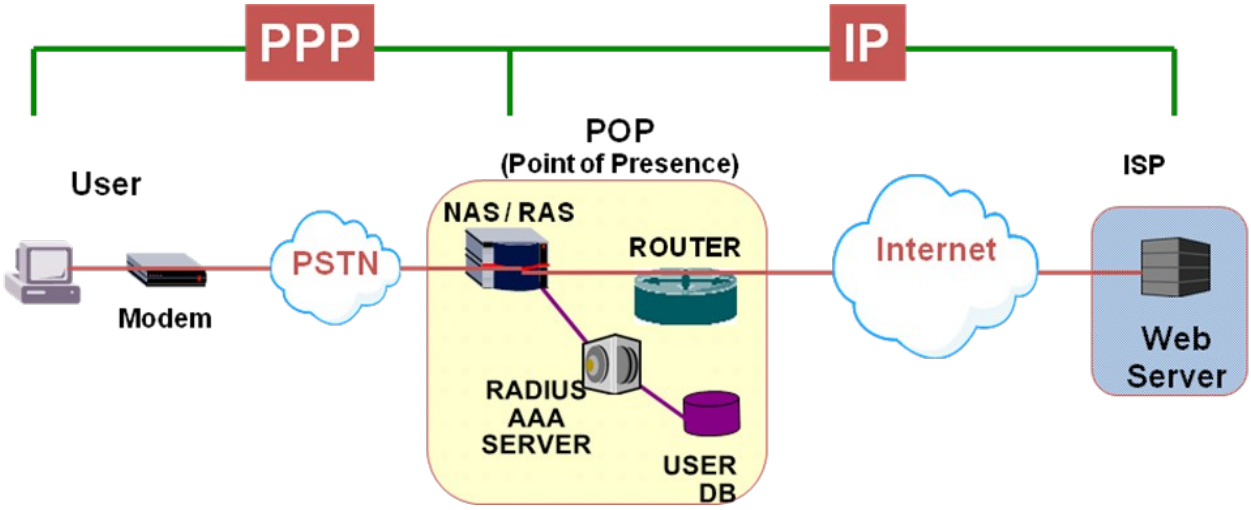
love is blue → 03d4ad6e7fee3f54eb46b5ccde58249c

love is Blue → 82b76f8eeb4a91aa640f9a23016c7b1c

1.6 Các ứng dụng công nghệ và mô hình triển khai Radius hiện nay :

1.6.1 Ứng dụng công nghệ :

- *Xác thực chuyển mạch Internet:* Trước đây việc truy cập internet hầu hết được thực hiện theo hình thức quay số (dial up). NAS và Radius Server làm cầu nối trung gian thực hiện xác thực tài khoản người dùng và cấp quyền truy cập internet.

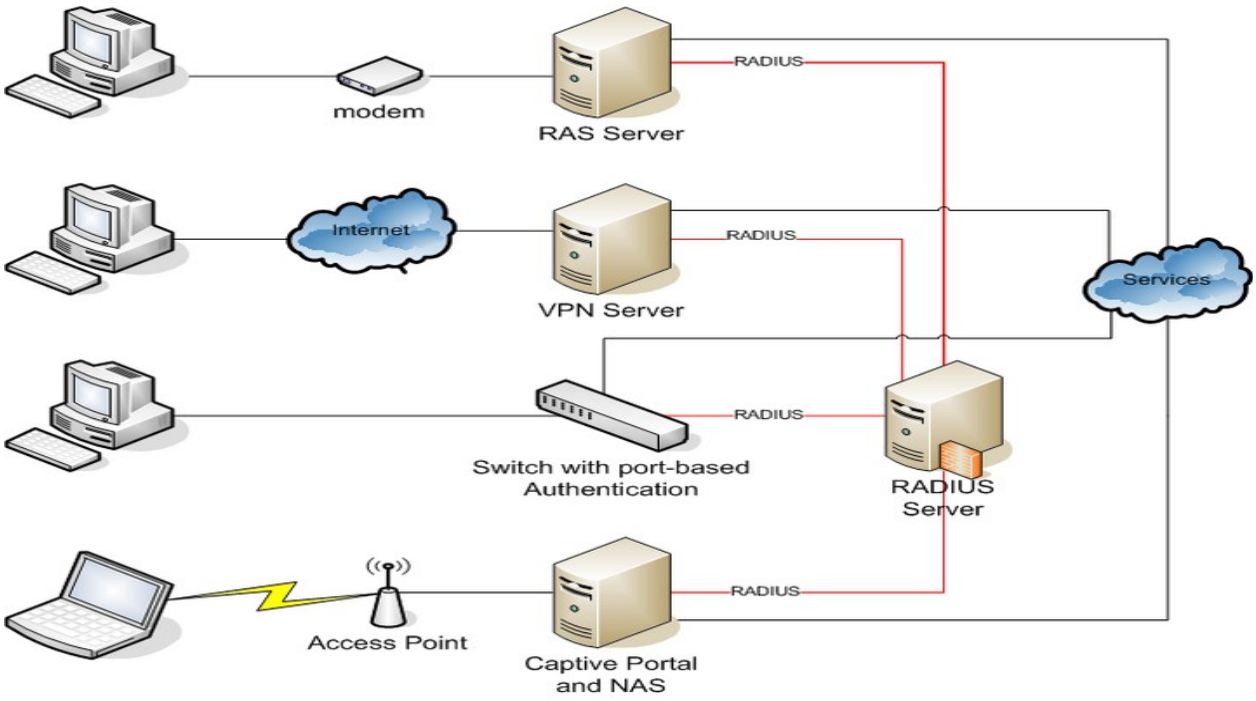


HÌNH 1.7 SƠ ĐỒ XÁC THỰC CHUYỂN MẠCH INTERNET

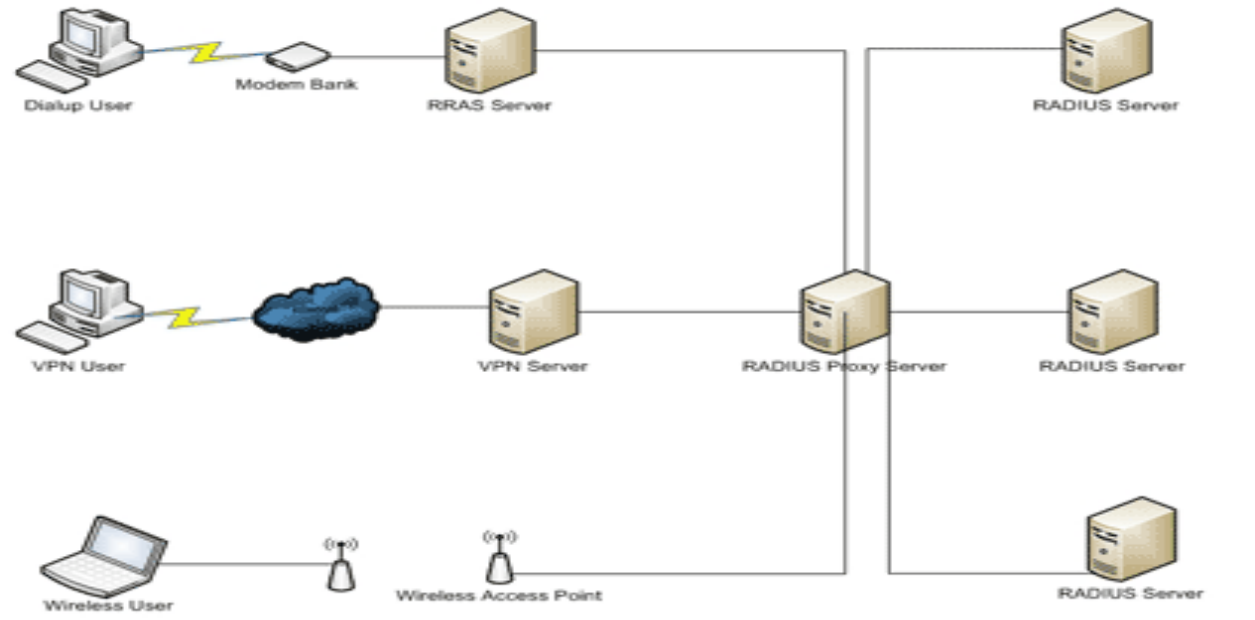
- *Truy cập DSL :* Mặc dù hiện nay việc kết nối internet bằng Dial-up đã trở nên lạc hậu và thay thế bằng ADSL nhưng với kiến trúc AAA cùng các tính năng hỗ trợ (xác thực tập trung, cấp quyền hoặc ngăn chặn các yêu cầu, và kế toán, tài liệu hóa trạng thái người dùng trong một phiên làm việc) Radius vẫn là giao thức số một với các nhà cung cấp phân phối mạng internet (ISP).
- *VPN (Vitual Private Network):* là một mạng dành riêng để kết nối các máy tính của các công ty, tập đoàn hay các tổ chức với nhau thông qua mạng Internet công cộng. Công nghệ **VPN** chỉ rõ 3 yêu cầu cơ bản:
  - Cung cấp truy nhập từ xa tới tài nguyên của tổ chức mọi lúc, mọi nơi.
  - Kết nối các chi nhánh văn phòng với nhau.
  - Kiểm soát truy nhập của khách hàng, nhà cung cấp và các thực thể bên ngoài tới những tài nguyên của tổ chức.

  Mặc dù VPN có các cơ chế bảo mật riêng nhưng để tăng cường và đảm bảo về bảo mật người ta thường sử dụng kèm thêm các giao thức bảo mật để tiến hành xác thực, điều khiển truy cập đối với các VPN client thực hiện truy cập từ xa vào hệ thống công ty. Ngoài ra còn có một lợi ích nữa là có thể quản lý account người dùng, account dùng để connect VPN được khởi tạo trong domain Win2k3, nên sẽ giúp Admin quản lý thuận tiện hơn.
- *Wireless Network:* với tính chất là một giao thức mở rộng, Radius đã không ngừng cải thiện và mở rộng, và hiện nay có khả năng xác thực cho cả các kết nối mạng không dây. Radius hỗ trợ các chuẩn 802.1X, 802.11i, 802.16e.

1.6.2 Một số mô hình triển khai Radius :



HÌNH 1.8 MÔ HÌNH TRIỂN KHAI RADIUS



HÌNH 1.9 MÔ HÌNH TRIỂN KHAI RADIUS CÓ PROXY

1.7 Ưu điểm và khuyết điểm :

1.7.1 Ưu điểm :

- Được xây dựng dựa trên mô hình Client/Server. Trong đó NAS đóng vai trò là một Radius Client chịu trách nhiệm chuyển thông tin từ người dùng lên Radius Server cũng như làm cầu nối trung gian giúp Server phản hồi lại cho người dùng.
- Là giao thức bảo mật mạng hỗ trợ khá nhiều dịch vụ an ninh, cũng như các thuật toán bảo mật.
- Các giao dịch giữa Client/Server đều được xác thực thông qua mã chia sẻ bí mật share secret và chỉ thực hiện trong mạng nội bộ. Đồng thời password của người dùng cũng được mã hóa trong các thông điệp giảm thiểu nguy cơ bị tấn công.
- Có cơ chế xác thực linh hoạt. Radius hỗ trợ khá nhiều phương thức chứng thực cho người dùng (PPP PAP, PPP CHAP, Unix Login, PPTP L2TP(VPN), và nhiều phương thức khác ….)
- Với bản chất là giao thức mở rộng nên giao thức không ngừng mở rộng và phát triển để trở nên hoàn thiện hơn, phục vụ cho hầu hết tất cả thiết kế mạng. Đặc biệt là đối với các hệ thống lớn

1.7.2 Khuyết điểm :

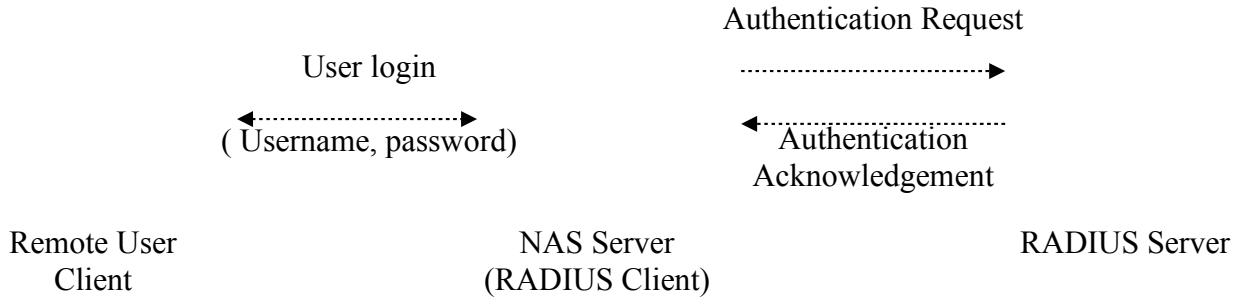
Mặc dù là một giao thức bảo mật mạng đa năng, linh hoạt nhưng và mặc dù được cải thiện mở rộng khá nhiều nhưng Radius vẫn tồn đọng một số hạn chế :

- Vấn đề bảo mật ở một số thiết kế hệ thống chưa được hoàn hảo, đặc biệt là với các hệ thống xây dựng nhiều proxy server Radius. Mỗi bước nhảy thông qua proxy thông tin của user sẽ được giải mã rồi tiếp tục mã hóa để chuyển tiếp. Những thông tin nhạy cảm của người dùng được chuyển tiếp qua nhiều cầu trung gian nguy cơ bị tấn cộng càng cao hơn mặc dù là thông tin được mã hóa hoặc ẩn dấu trong các thông điệp.
- Chưa có cơ chế hủy bỏ (recalling) và thu hồi tài nguyên (deallocating resource) khi tiến hành xác thực. Trong trường hợp hệ thống xây dựng nhiều proxy server. Sau khi server đầu nhận được thông tin xác thực để chuyển tiếp đến các server trung gian. Một số trường hợp xảy ra như là hết thời gian truy cập trong ngày, thì Radius không hỗ trợ từ chối hoặc ngắt kết nối đối với phiên truy cập đó.
- Mặc dù tính phi trạng thái (stateless) hỗ trợ cho quá trình vận chuyển gói tin trong hệ thống trở nên nhanh chóng và giảm độ phức tạp nhưng đây lại chính là một hạn chế của Radius. Với tính chất này hệ thống sẽ không ghi nhận lại bất cứ các thiết lập cấu hình, thông tin giao dịch, hoặc dữ liệu nào khác cho phiên tiếp theo. Điều này làm phức tạp trong việc tìm các giải pháp quản lý tài nguyên và thời gian tự động thoát.
- Việc mở rộng cho RADIUS như một con dao 2 lưỡi, có thể làm giảm hiệu suất và dữ liệu bị mất khi được sử dụng trong các hệ thống quy mô lớn, một phần bởi vì nó không bao gồm các quy định kiểm soát tắc nghẽn trong quá trình hoạt động và sử dụng.

CHƯƠNG 2 : CẤU TRÚC, NGUYÊN LÝ HOẠT ĐỘNG VÀ THUẬT TOÁN ÁP DỤNG

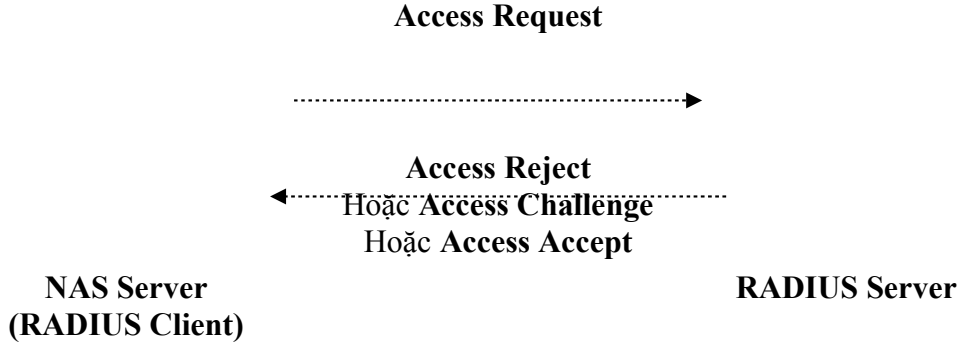
2.1 Nguyên lý hoạt động chung :

2.1.1 Xác thực và ủy quyền:



Hinh 2.1 MÔ HÌNH TIẾN HÀNH XÁC THỰC CHO NGƯỜI DÙNG

Khi một người dùng cần truy cập vào hệ thống để sử dụng các dịch vụ hay tài nguyên thì cần phải thông qua bước xác thực thông tin cá nhân. Việc xác thực này được thực hiện khá đơn giản thông qua một cửa sổ đăng nhập, là nơi mà người dùng phải nhập username và password vào. Tiếp theo đó người dùng sẽ lựa chọn một trong những giao thức phù hợp (chẳng hạn như PPP PAP, PPP CHAP, PPTP, L2TP …) để gửi qua internet các gói dữ liệu chứa các thông tin xác thực này đến hệ thống. Khi NAS Server (đảm nhiệm vai trò là một Radius Client) nhận được những thông tin xác thực do người dùng gửi đến, sẽ tiến hành sử dụng giao thức Radius để bắt đầu quá trình xác thực cho người dùng với Radius Server. Radius Client sẽ tạo ra một yêu cầu truy cập (Access-Request) bao gồm các một số thuộc tính quan trọng như : username, password của người dùng, ID nhận diện của Radius client (NAS ID) cũng như thông số cổng (PortID) mà người dùng truy cập vào (thông số port ở đây không phải là các port socket). Thộc tính mật khẩu lúc này sẽ được ẩn thông qua phương pháp hàm băm MD5.



Hình 2.2 SƠ ĐỒ MINH HỌA VIỆC GIAO TIẾP GIỮA RADIUS CLIENT VÀ RADIUS SERVER TRONG QUÁ TRÌNH XÁC THỰC – CẤP QUYỀN

Vì giao thức Radius dùng UDP làm phương tiện trung chuyển các gói tin vì vậy rất có khả năng các gói tin không đến được đích. Do đó nếu server không có phản hồi sau khoảng thời gian đã quy ước thì yêu cầu được gửi lại. Trong trường hợp hệ thống có nhiều Radius server thì Radius client sẽ chuyển tiếp yêu cầu đến các server dự phòng này nếu server chính hư hỏng, không hoạt động.

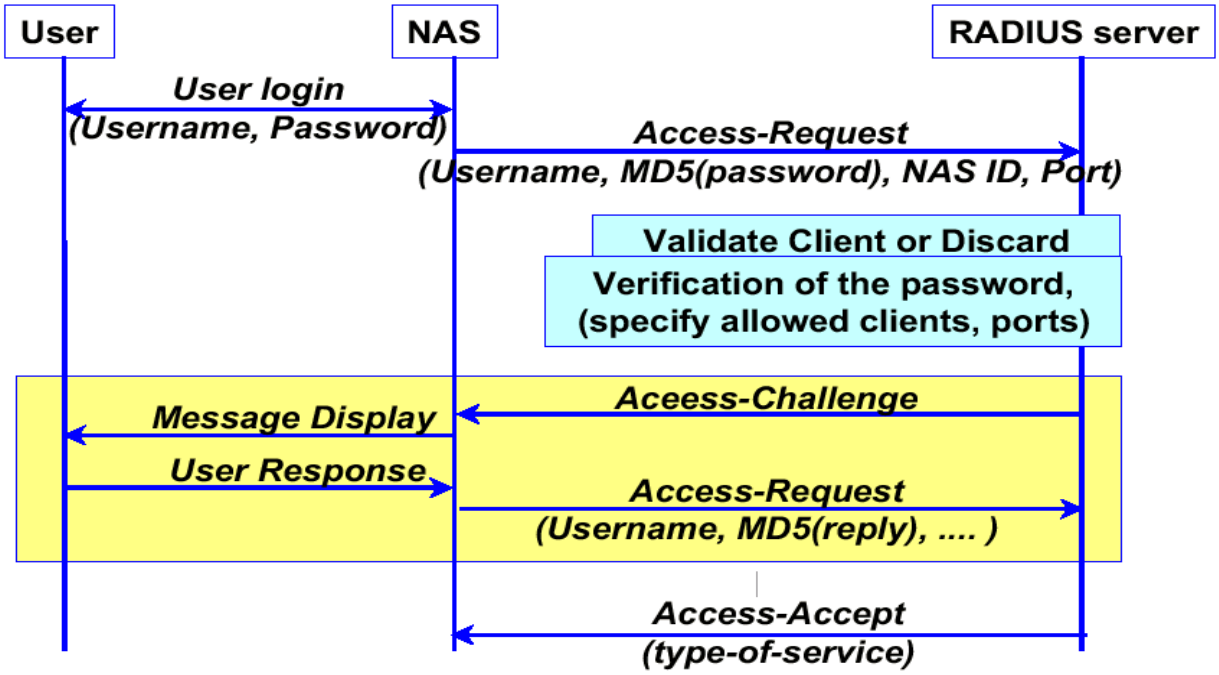
Khi Radius server nhận được yêu cầu gửi đến, server tiến hành xác minh và kiểm tra user truy cập đến. Tuy nhiên nếu một Access Request được gửi đến từ một Radius Client mà không được tiến hành share secret với Radius server thì Radius sẽ tự động loại bỏ mà không phản hồi với các yêu cầu này (slient discard). Nếu yêu cầu được gửi đến từ một Radius client hợp lệ (đã share secret) , Radius Server tiến hành kiểm tra cơ sở dữ liệu về thông tin của người dùng này. Hệ thống sẽ tiến hành kiểm tra các thông tin do người dùng gửi đến so với các thông tin lấy từ cơ sở dữ liệu. Nếu thông tin hoàn toàn trùng khớp hệ thống sẽ cấp quyền truy cập cho user đó. Việc kiểm tra này bao gồm cả việc xác minh trường password cũng như các tài nguyên mà người dùng được phép truy cập.

Ngoài ra, nếu không đủ thẩm quyền giải quyết yêu cầu cho user, Radius Server lúc này sẽ đóng vai trò như một client , tạo và gửi yêu cầu tham chiếu đến các Server radius khác. Quá trình tiếp tục lặp lại cho đến khi giải quyết được yêu cầu cầu của người dùng. Trường hợp hệ thống có sử dụng Proxy Radius Server thì trong các Access- Request thường bao gồm trường Proxy-State. Hệ thống sẽ sao chép thông số của trường này và luôn gửi kèm theo trong các phản hồi mà không chỉnh sửa gì.

Đối với trường hợp các điều kiện không hợp lệ, Radius Server tiến hành gửi trả một phản hồi Access-Reject để thông báo với người dùng là yêu cầu gửi đến không hợp lệ. Ngoài ra trong phản hồi Access-Reject có thể chứa đoạn text hiển thị thông báo cho người dùng và không chứa bất kì trường thông tin nào (ngoại trừ Proxy State đã nói ở trên).

Trường hợp đối với điều kiện hợp lệ, server có thể sẽ gửi một yêu cầu Access-Challenge và yêu cầu người dùng phải phản hồi. Yêu cầu Access Challenge có thể chứa một thông điệp văn bản sẽ hiển thị bên máy người dùng. Nếu máy người dùng nhận được yêu cầu và hỗ trợ cơ chế thách thức / phản hồi (challenge/response) thì đoạn thông điệp sẽ hiển thị và yêu cầu người dùng phản hồi lại hệ thống. Sau khi nhận được các phản hồi từ phía người dùng, Radius client tiến hành gửi lại Access Request ban đầu với một số ID khác và trường user-password được thay thế bằng thông tin phản hồi của người dùng (đã được mã hóa).Đối với yêu cầu gửi lại này, tùy theo tình huống mà Radius Server sẽ gửi phản hồi Access Reject để từ chối, Access Accept để đồng ý hoặc một Access Challenge khác.

Nếu bên người dùng đáp ứng tất cả các điều kiện yêu cầu, Radius Server sẽ phản hồi Access-Accept trong đó sẽ chứa các giá trị cấu hình của người dùng. Những giá trị cấu hình này bao gồm loại dịch vụ (type of service) chẳng hạn như PPP, SLIP, Login User(người dùng dăng nhập) và tất cả các giá trị cần thiết mà dịch vụ, tài nguyên của người dùng yêu cầu.



HÌNH 2.3 SƠ ĐỒ MINH HỌA VIỆC XÁC THỰC – CẤP QUYỀN THÀNH CÔNG CÓ SỬ DỤNG CHALLENGE

❖ Cơ chế thách thức/ phản hồi (challenge/response):

Trong cơ chế xác thực này, người dùng sẽ được cung cấp một con số ngẫu nhiên, không đoán trước được và được thánh thức mã hóa con số này và gửi lên hệ hệ thống. Đối với người dùng xác thực được phần mềm hoặc công cụ hỗ trợ tạo điều kiện thực hiện tính toán và phản hồi một cách dễ dàng. Nhưng đối với người dùng trái phép thì không có mã chia sẻ bí mật , nên chỉ có thể đoán mò lời phản hồi. Dựa vào đây server có thể xác nhận đâu là yêu cầu từ người dùng hợp lệ và không hợp lệ. Người dùng sẽ nhập con số Challenge này vào phần mềm hoặc thiết bị hỗ trợ, các thiết bị hay phần mềm sẽ tự tính toán và gửi phản hồi cho hệ thống. Và NAS đóng vai trò như một Radius Client sẽ chuyển tiếp phản hồi này đến với Radius Server dưới một yêu cầu Access Request.

Ví dụ như theo hình 2.3 user đăng nhập vào hệ thống, nhập vào user name và password. NAS chuyển tiếp lên Server trong Access Request bao gồm một số trường thông tin như Username, User-password (là chuỗi cố định tương tự "challenge" hoặc có thể không có), NAS ID, NAS port. Server sẽ phản hồi một Access-Challenge bao gồm thông điệp hiển thị và một chuỗi challage (giả sử là 12345678). Khi máy người dùng nhận được , màn hình hiển thị thông điệp chuỗi challenge và yêu cầu người dùng nhập lại. Sau khi người dùng đã nhập xong, phần mềm hoặc thiết bị tự tính toán và phản hồi lại cho hệ thống. Khi NAS nhận được phản hồi từ phía người dùng sẽ gửi một Access Request có ID number khác chứa các thông tin của yêu cầu trước đó như Username, NAS ID, NAS port và User-password (lúc này trường này đã được thay

thế bằng chuỗi kí tự đã được mã hóa mà người dùng đã nhập trước đó). Server sẽ tiến hành tự tính toán và so sánh với phản hồi nhận được để thực hiện bước tiếp theo.

❖ PAP và CHAP :

Nếu hệ thống dùng PAP thì NAS sử dụng PAP ID và password thay thế cho trường Username và Password trong Access-Request. Ngoài ra còn một số trường như Service-Type = Framed-User and Framed-Protocol = PPP để gợi ý cho RADIUS server dịch vị dư kiến là PPP.

Đối với CHAP , NAS tạo ra một giá trị challenge ngẫu nhiên (độ dài khoảng 16octet) và người dùng sẽ phản hồi lại một giá trị là CHAP Response cùng với CHAP ID và CHAP username. NAS sẽ gửi trong Access Request có giá trị user name là CHAP Username và Password chính là CHAP Response. Giá trị ngẫu nhiên được tạo ra ban đầu được đặt trong trường Authenticator (Request Authenticator). Tương tự như PAP , CHAP cũng chứa các trường Service-Type = Framed-User and Framed-Protocol = PPP để gợi ý cho RADIUS server dịch vụ dự kiến là PPP.

❖ Trong quá trình xác thực & cấp quyền Radius thực hiện lần lượt các công việc sau :

- Decrypt password của user (do encrypt trong quá trình vận chuyển) bằng cách sử dụng share secret key (cả Client và Server đều biết mã này)

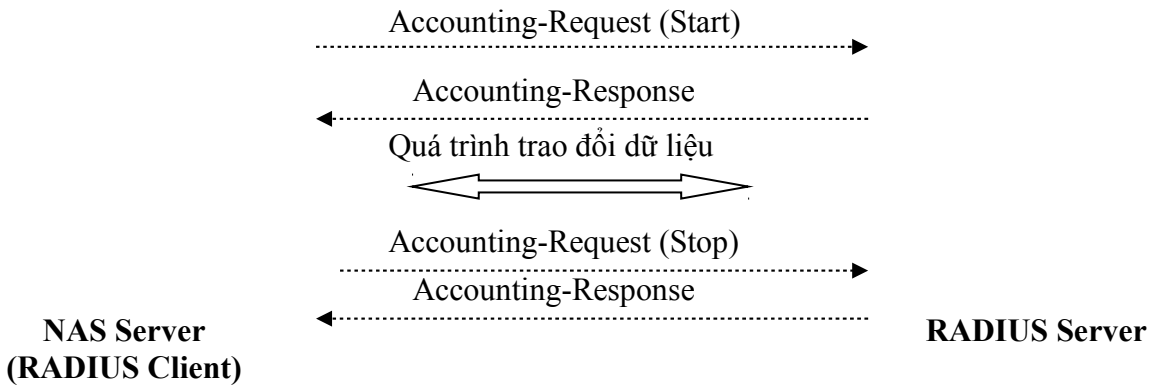
- Tìm kiếm thông tin tài khoản của user trong CSDL.
  - Plain text file
  - Hệ quản tri CSDL SQL
  - LDAP server
  - /etc/passwd trong HDH Linux
  - Active Directory của HDH Window
  - etc

- Xác thực cho user

- Kiểm tra các tùy chọn (check-items)
  - Loại kết nối (POTS, ISDN, ADSL, cable, UMTS, etc.)
  - Thời gian trong ngày
  - Calling number, called number
  - etc.

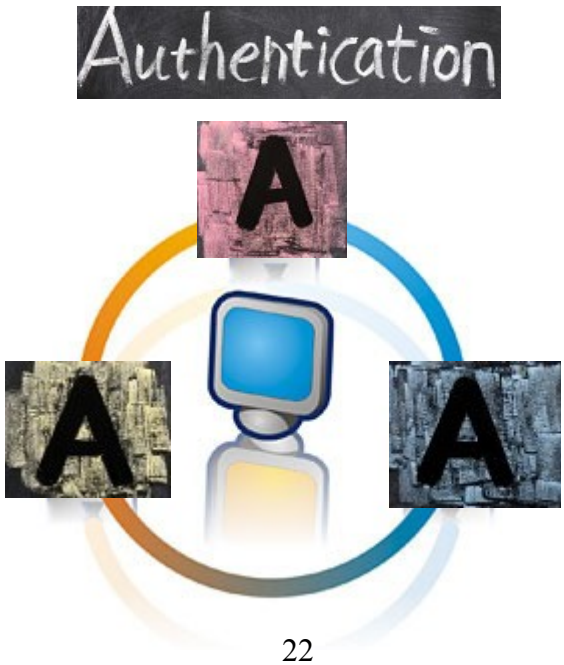
- Gửi Access-Accept/Reject đến NAS với những thuộc tính phù hợp cho phiên làm việc (reply-items)
  - Idle and session timeout
  - Bộ lọc Ip cho người dùng
  - Chỉ định IP gán cho người dùng
  - Đối với ISDN, số lương kênh phát tối đa (MLPPP)
  - etc.

2.1.2 Kế toán:

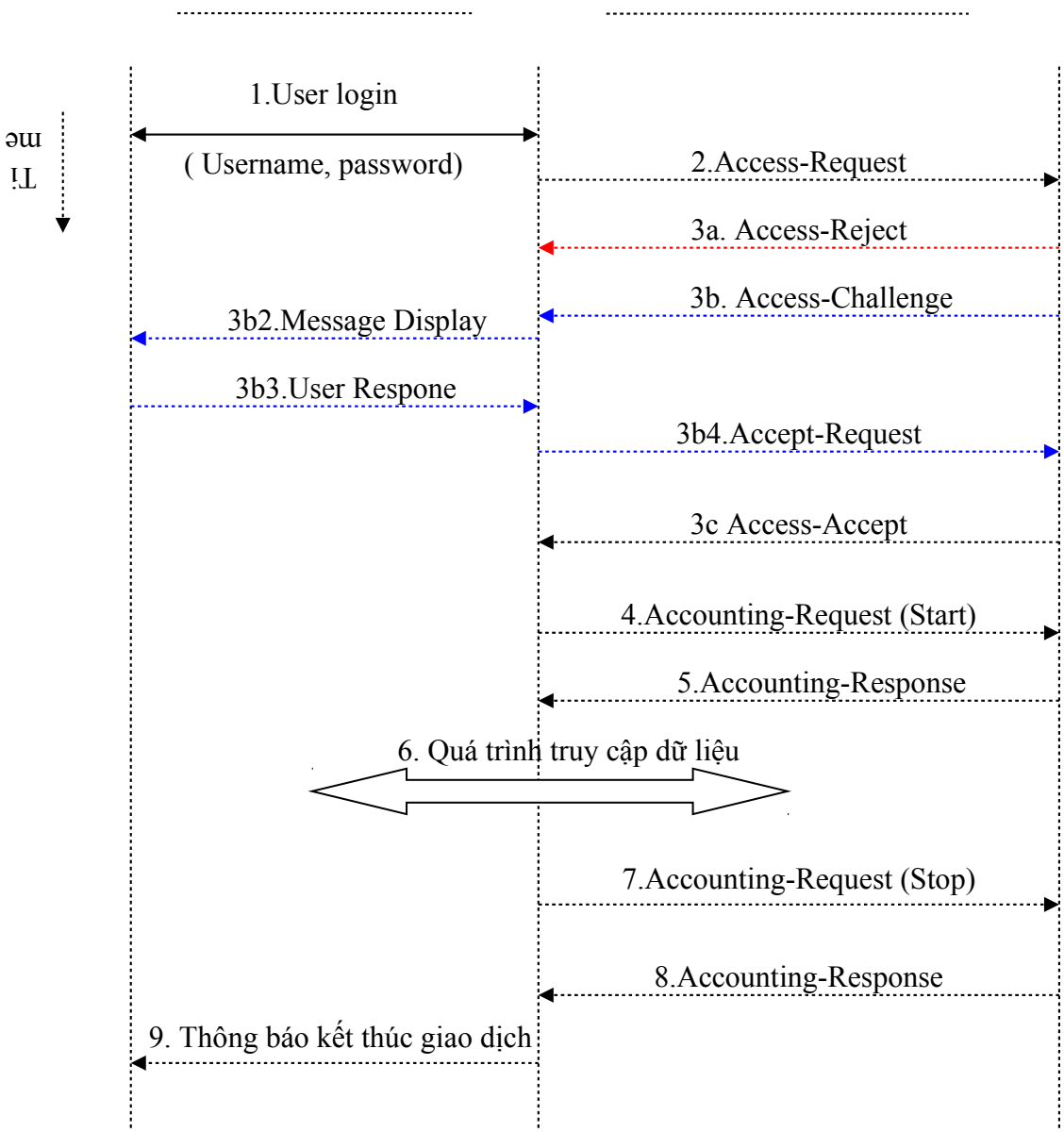


HÌNH 2.4 SƠ ĐỒ MÔ HÌNH TRAO ĐỔI THÔNG ĐIỆP TRONG KẾ TOÁN

Nếu người dùng được cấu hình sử dụng dịch vụ kế toán (Accounting), khi bắt đầu cung cấp dịch vụ, Radius Client sẽ tiến hành gửi một thông yêu cầu Accounting Request (Start) mô tả các loại hình dịch vụ mà người dùng được cung cấp và tiến hành gửi cho Radius Server. Khi nhận được yêu cầu này Radius Sever sẽ phản hồi một thông điệp Accounting-Response thừa nhận đã nhận yêu cầu ở trên. Khi kết thúc phiên làm việc , client cũng tiến hành gửi yêu cầu Accounting Request (Stop) bao gồm mô tả về các loại hình dịch vụ sử dụng cũng như một số tùy chọn thống kê sử dụng như thời gian sử dụng, lưu lượng dữ liệu truyền tải, tốc độ truy cập trung bình cùng một số chi tiết khác …Và sau đó cũng sẽ phản hồi cho client biết là đã nhận được yêu cầu. Trong trường hợp Radius Server không lưu lại được gói tin gửi đến thì sẽ không phản hồi trở lại cho client.



Remote User Client NAS Server (RADIUS Client) RADIUS Server



HÌNH 2.5 SƠ ĐỒ NGUYÊN LÝ LÀM VIỆC CỦA GIAO THỨC RADIUS

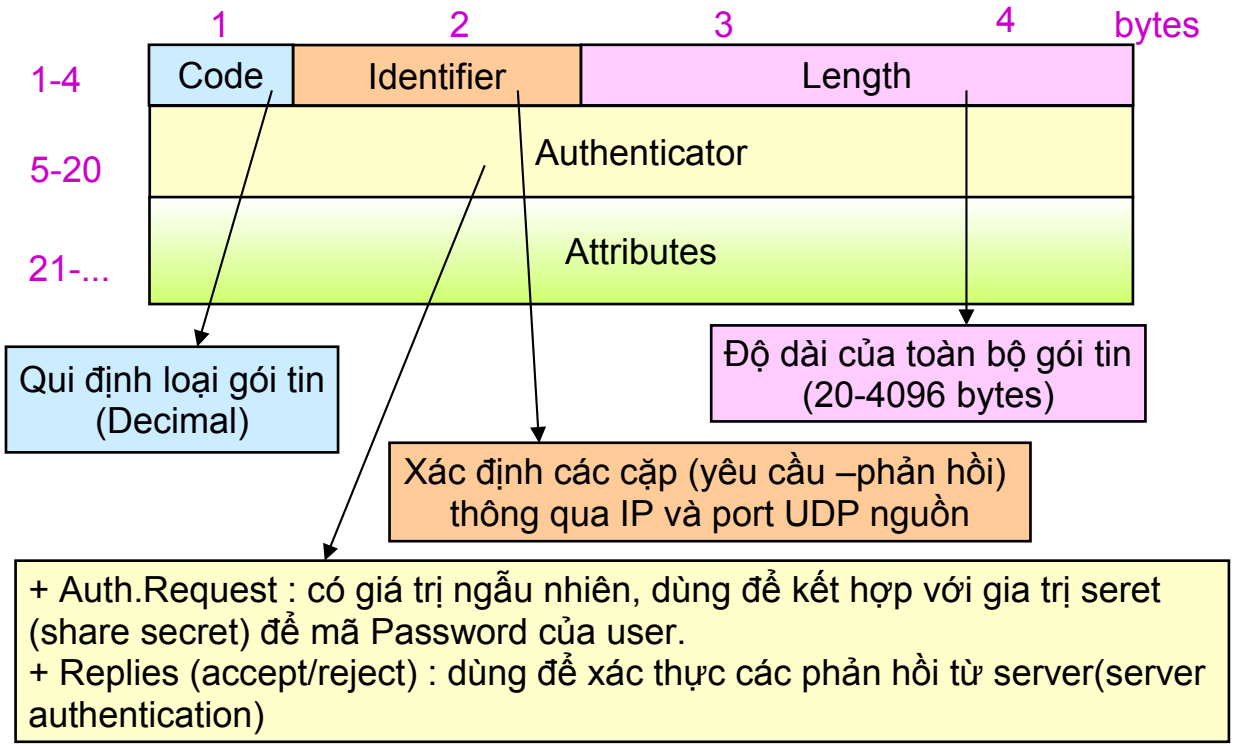
* Các bước thực hiện xác thực , cấp quyền , kế toán trong giao thức Radius :
B1 : User tiến hành nhập Username & Password đăng nhập vào hệ thống
B2 : NAS(Radius Server) nhận được thông tin của người dùng tiến hành gửi thông điệp Access-Request để yêu cầu Server chứng thực cho người dùng.

B3: Server tiến hành kiểm tra thông tin của người dùng để xác định bước kế tiếp

- Nếu các thông tin không hợp lệ, Server sẽ phản hồi lại Acess-Reject để từ chối yêu cầu
- Để đảm bảo Server gửi một yêu cầu Access-Challenge yêu cầu người dùng phản hồi. Người dùng tiến hành nhập mã phản hồi và gửi lên lại hệ thống. NAS chuyển tiếp phản hồi của người dùng lên Server. Server lúc này có thể từ chối bằng Access-Reject, chấp nhập bằng Access-Accept hoặc gửi một Access-Challenger khác.
- Nếu thông tin hợp lệ Server sẽ phản hồi Access-Accept.

B4: Radius Client gửi yêu cầu thực hiện dịch vụ Accounting bằng yêu cầu Accounting-Request (Start)
B5: Server phản hồi đã nhận được yêu cầu bằng Accounting-Response
B6: Quá trình truy cập dữ liệu của user
B7: Khi người dùng ngắt kết nối, Radius Client gửi một yêu cầu ngưng dịch vụ Accounting bằng yêu cầu Accounting-Request (Stop).
B8: Server sau khi đã nhận được yêu cầu sẽ phản hồi bằng Accounting-Response
B9: Thông báo cho người dùng phiên làm việc kết thúc.

2.2 Cấu trúc gói tin:



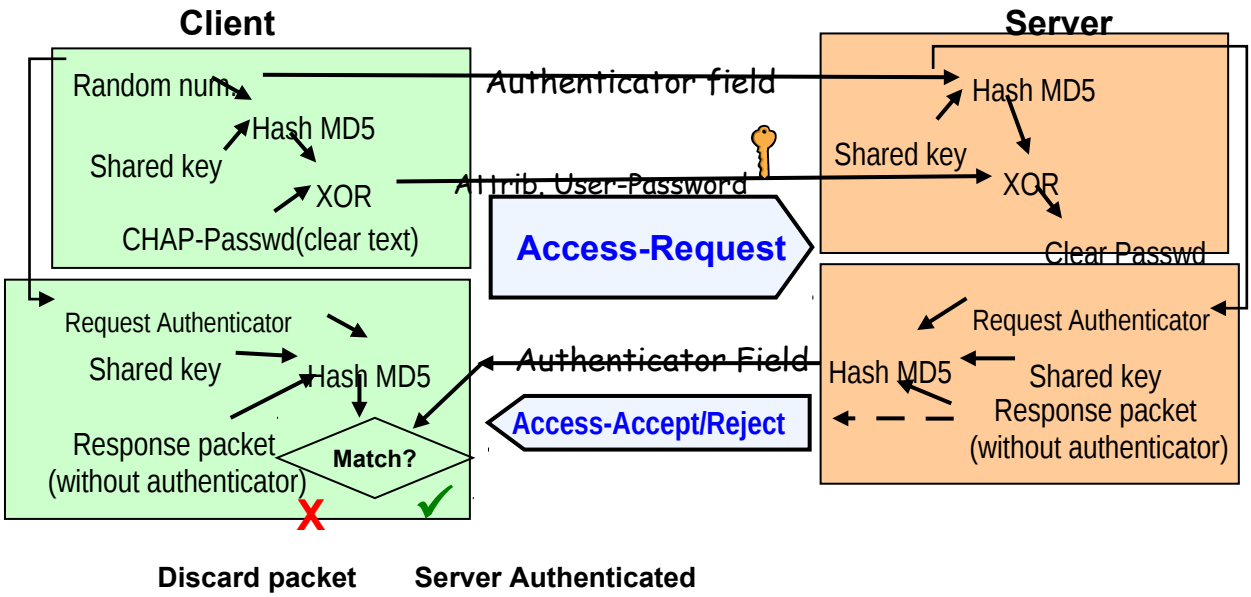
2.2.1 Phân loại gói tin:

Người ta dựa vào phần code (chiếm 1 byte) để phân loại gói tin

Code	Tên	Mô tả
1	Access-Request	Yêu cầu chứng thực do NAS (Radius Client) gửi cho Radius Server
2	Access-Accept	Phản hồi từ server đến NAS đã chấp nhận phiên làm việc của user.
3	Access-Reject	Phản hồi từ server đến NAS từ chối phiên làm việc của user
11	Access-Challenge	Yêu cầu gửi từ Server cho NAS để yêu cầu User cung cấp thêm thông tin bổ sung
4	Accounting-Request	NAS gửi thông tin kế toán cho server
5	Accounting-Response	Server xác nhận đã nhận được gói tin bằng ACKs

BẢNG PHÂN LOẠI CÁC GÓI TIN THEO MÃ CODE

2.2.2 Trường Authenticator :



HÌNH 2.6 CHỨC NĂNG CỦA TRƯỜNG AUTHENTICATOR TRONG XÁC THỰC VÀ CẤP QUYỀN

Theo như hình 2.6, nhiệm vụ của trường Authenticator nhằm phục vụ cho 2 mục đích, tùy thuộc vào đó là một yêu cầu (Access-Request) hay là một phản hồi (Access Reject/Accept).

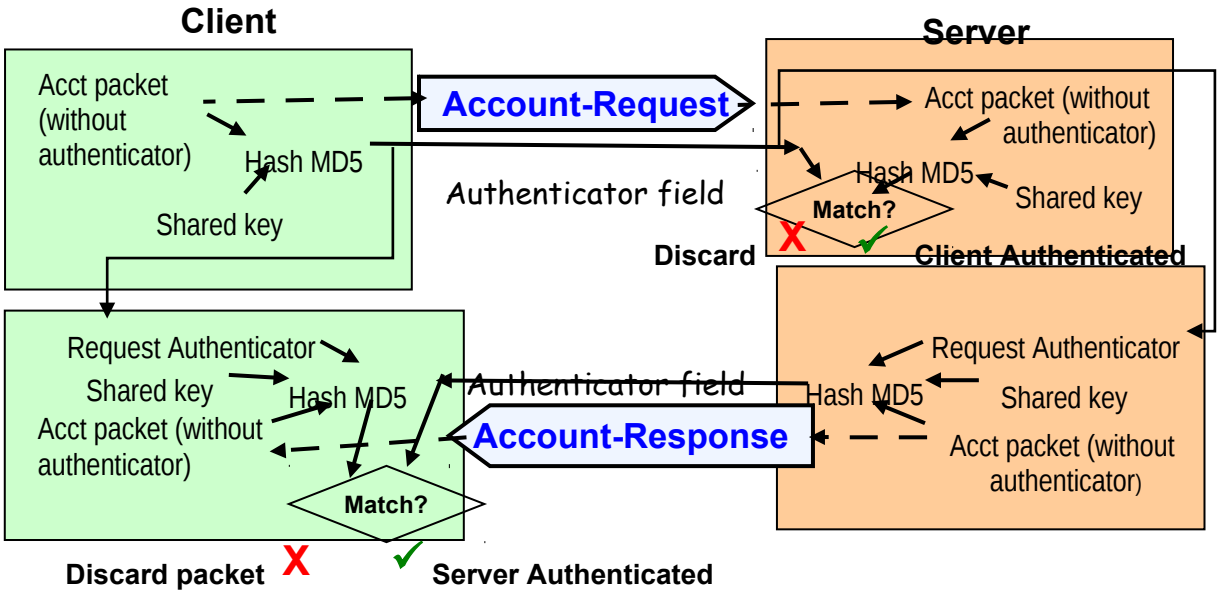
- Mã hóa trường thông tin Password của user (phục vụ dịch vụ bảo mật thông tin)
- Xác thực thông tin của User (phục vụ dịch vụ xác thực)

Sau khi nhận được thông tin của người dùng gửi đến, Radius Client gửi một Access-Request đến Radius Server xin xác thực cho người dùng. Trường

Authenticator trong gói tin Access-Request được khởi tạo bằng một con số ngẫu nhiên (16 octet), được gọi là Request Authenticator. Trường User-Password (Chap-Password) được mã hóa bằng Request Authenticator cùng mã bí mật chia sẻ (share secret) với thuật toán hàm băm MD5.

Khi server nhận được gói tin Access-Request do Radius Client gửi đến sẽ tiến hành giải mã Password của người dùng và kiểm tra trong CSDL để quyết định bước tiếp theo. Ở bước tiếp theo này Radius Server sẽ phản hồi lại Radius Client một trong 3 lựa chọn (Access-Accept nếu hợp lệ, Access Reject nếu không hợp lệ hoặc một Access-Challenge để kiểm tra nếu cần). Lúc này trường Authenticator sẽ được tính toán lại bằng cách dùng hàm băm **MD5(Code + Identifier + Length + RequestAuthenticator + Attributes + Secret(share secret).** Lúc này, trường Authenticator trong các thông điệp phản hồi này được gọi là Response Authenticator. Mục đích chính của Response Authenticator dùng để xác thực cho các thông điệp gửi đến chính là của Radius Server (phục vụ mục đích xác thực). Ngoài ra kết hợp với mã số bí mật secret (share secret) để kiểm tra tính toàn vẹn dữ liệu trong quá trình vận chuyển trong mạng.

Khi Radius Client nhận được các thông điệp phản hồi từ phía Server, sẽ tiến hành kiểm tra xác thực xem gói tin đó có đúng của Server gửi đến không. Nếu không trùng khớp sẽ tự động loại bỏ gói tin đó.



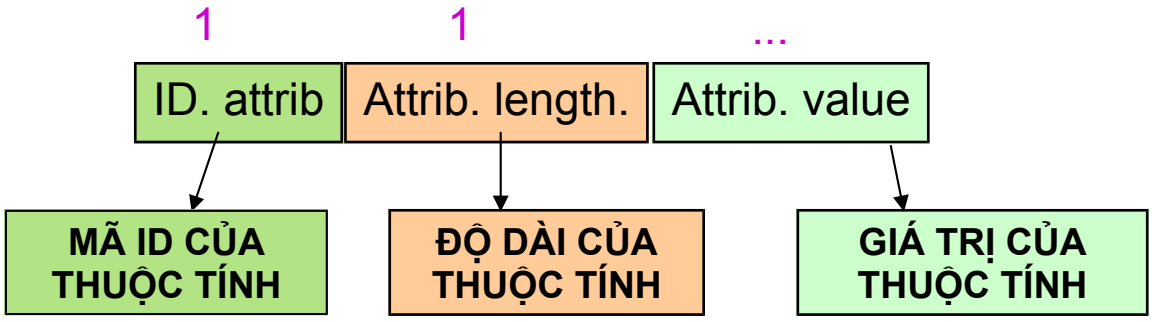
HÌNH 2.7 SƠ ĐỒ MINH HỌA CHỨC NĂNG TRƯỜNG AUTHENTICATOR TRONG ACCOUNTING

Đối với tính năng Accounting , trường Authenticator chỉ phục vụ một chức năng duy nhất đó là xác thực cho cả 2 phía Client và Server, nghĩa là hoạt động như một chữ kí điện tử, phục vụ cho dịch vụ chống chối bỏ giữa 2 bên tham gia. Đối với yêu cầu Accounting Request, trường Request Authenticator không còn là một con số

ngẫu nhiên nữa mà sẽ được tính toán bằng **MD5 (Code + Identifier + Length + 16 zero octets + Attributes + Secret).**

Khi Radius Server nhận được gói tin Accouting Request, sẽ tiến hành kiểm tra xem đó có phải là do chính Radius Client gửi đến hay không (bằng cách tính toán dự trên hàm băm MD5 và mã secret). Nếu hợp lệ server sẽ phản hồi bằng thông điệp Accounting Response để thừa nhận đã nhận được yêu cầu do Radius Client gửi đến. Lúc này bên Client cũng thực hiện tương tự để xác nhận chiều ngược lại. Nếu hợp lệ thì cả hai đã xác thực lẫn nhau trong giao dịch. Trường Authenticator trong thông điệp phản hồi tính bằng **MD5(Code + Identifier + Length + RequestAuthenticator + Attributes + Secret).**

2.3 Ý nghĩa một số trường thuộc tính:



- ***User-Name (1)*** – Length : 3 octet trở lên – Kiểu : String

Thường xuất hiện: Access-Request & Accounting-Request

Ý nghĩa :Phân biệt giữa các khách hàng tham gia truy cập vào hệ thống trên mạng. Ngoài ra các máy chủ RADIUS cũng có thể sử dụng một tên người dùng để xác định hành vi thích hợp cho một giao dịch.

- ***User-Password (2)*** – Length : 18 -130 octet – Kiểu : String

Chỉ xuất hiện: Access-Request .

Thuộc tính này được thiết kế để mang thông tin xác thực mà người dùng cung cấp để truy cập các dịch vụ mạng. Chủ yếu, nội dung của giá trị này sẽ là một mật khẩu mã hóa, nhưng đôi khi nó có thể là phản ứng từ một gói tin Access-Challenge gửi cho khách hàng từ các máy chủ RADIUS.

- ***CHAP-Password (3)*** – Length: 19 octet – Kiểu : String

Chỉ xuất hiện: Access-Request

Mục đích đầu tiên là cho biết giao thức xác thực người dùng sử dụng là CHAP, sẽ được sử dụng trong quá trình giao dịch. Chức năng tương tự như User-Password.

- ***CHAP-Challenge (60)*** – Length : 7 octet trở lên – Kiểu : chuỗi

Xuất hiện trong : Access-Request

Phục vụ cho việc xác thực user bằng cơ chế xác thực CHAP. CHAP challenge được tính bằng MD5 (Password + chuỗi challenge)

- ***NAS-Identifier(32)*** – Length : 3 octet trở lên – Kiểu : chuỗi

Xuất hiện trong : Access-Request

Thuộc tính này xác định các NAS mà xây dựng các gói tin Access-Request. Thông thường, tên miền đầy đủ (FQDN) được sử dụng trong phần giá trị của thuộc tính này.

- ***NAS-IP-Address (4)*** – Length : 6 octet – Kiểu : IPAddress

Xuất hiện trong : Access-Request

Thuộc tính này xác định địa chỉ IP của thiết bị NAS có yêu cầu dịch vụ thay mặt cho các máy tính khách hàng nhưng RFC RADIUS không cho phép cả hai thuộc tính ***NAS-IP-Address*** và thuộc tính ***NAS-Identifier*** được sử dụng trong cùng một gói tin. Tuy nhiên, một trong hai phải có mặt trong gói tin bất kỳ.

- ***NAS-Port (5)*** – Length : 6 octet – Kiểu : Integer

Xuất hiện trong : Access-Request

Hiển thị port mà user đang kết nối với NAS (không phải là port dịch vụ kết nối – socket port), đại diện cho các port thực tế, hữu hình, vật lý trên các thiết bị kết nối. Nếu không có thì giá trị được đặt theo số ảo.

- ***NAS-Port-Type (61)*** – Length : 6 octet – Kiểu : ENUM (giá trị theo bảng định danh)

Xuất hiện trong : Access-Request

Cho biết kiểu NAS-port kết nối là gì . VD : Vitual(5)

Bảng các giá trị NAS-port-type

Value	Type of port
0	Asynchronous
1	Synchronous
2	ISDN Synchronous
3	ISDN Asynchronous V.120
4	ISDN Asynchronous V.110
5	Virtual
6	PIAFS

Value	Type of port
7	HDLC Clear Channel
8	X.25
9	X.75
10	G.3 Fax
11	SDSL
12	ADSL-CAP
13	ADSL-DMT
14	IDSL
15	Ethernet
16	XDSL
17	Cable
18	Wireless other
19	Wireless CCITT 802.11

- ***Proxy-State (61)*** – Length : 3 octet trở lên – Kiểu : String

Xuất hiện trong : Tất cả gói tin

Thuộc tính này được sử dụng khi một máy chủ RADIUS hoạt động như một proxy và nhu cầu để lưu thông tin về một yêu cầu nổi bật, chẳng hạn như địa chỉ IP, tên miền, hoặc các định danh duy nhất số nguyên

- ***Reply-Message (18)*** – Length : 3 octet trở lên – Kiểu : String

Xuất hiện trong : Tất cả gói tin ngoại trừ Access-Request

Giá trị này được sử dụng để cung cấp một tin nhắn cho khách hàng cho một gói tin khác. Thường được tìm thấy trong Acess-Accept, dùng để cung cấp một thông điệp chào mừng, một thông báo lỗi, hoặc các thông tin khác cho người sử dụng.

- ***Service-Type (6)*** – Length : 6 octet – Kiểu : ENUM

Xuất hiện trong : Access-Request, Access-Accept
Cho biết các loại dịch vụ mạng mà Radius client cung cấp cho user.
Bảng các giá trị

Value	Service type
1	Login
2	Framed
3	Callback Login
4	Callback Framed
5	Outbound
6	Administrative
7	NAS Prompt
8	Authenticate Only
9	Callback NAS Prompt
10	Call Check
11	Callback Administrative

- ***Acct-Status-Type (40)*** – Lenghth : 6 octet – Kiểu ENUM

Xuất hiện trong : Accouting-Request
Cho biết trạng thái kết nối hiện tại của dịch vụ Accouting
Bảng giá trị

Value	Status type
1	Start
2	Stop
3	Interim-Update
7	Accounting-On
8	Accounting-Off
9 -14	Reserved; used for tunnel accounting
15	Reserved; used for failed attempts

- ***Acct-Session-Id (44)*** – Length : từ 3 octet trở lên – Kiểu : chuỗi

Xuất hiện trong: Accouting-Request, Access Request
Dùng để xác định 1 phiên làm việc.

- ***Acct-Authentic (45)*** –Length : 6 octet – Kiểu : ENUM

Xuất hiện trong: Accouting-Request
Bảng giá trị

Value	Authentication method
1	RADIUS
2	Local
3	Remote

- ***Acct-Session-Time (46)*** –Length : 6 octet – Kiểu : Integer

Xuất hiện trong: Accouting-Request
Tính bằng giây, cho biết thời gian kết nối của người dùng.

- ***Acct-Terminate-Cause (49)*** –Length : 6 octet – Kiểu : ENUM

Xuất hiện trong: Accouting-Request

Cho biết nguyên nhân kết thúc một phiên làm việc của người dùng.
Bảng giá trị :

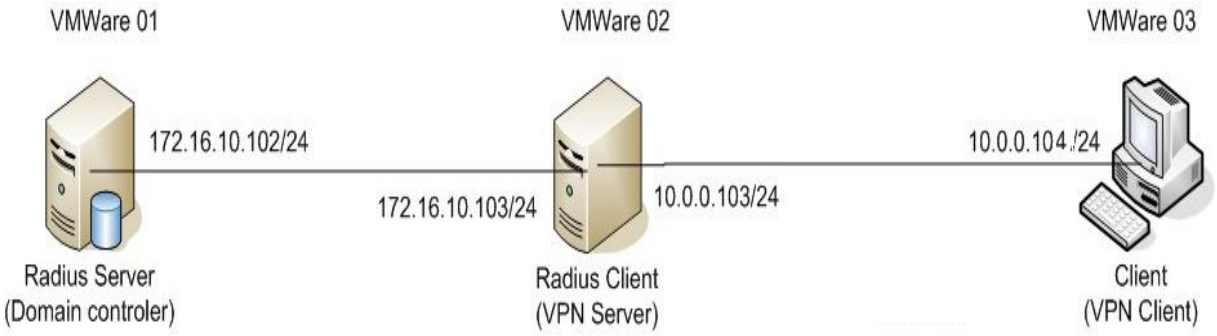
Value	Termination cause
1	User Request
2	Lost Carrier
3	Lost Service
4	Idle Timeout
5	Session Timeout
6	Admin Reset
7	Admin Reboot
8	Port Error
9	NAS Error
10	NAS Request
11	NAS Reboot
12	Port Unneeded
13	Port Preempted
14	Port Suspended
15	Service Unavailable
16	Callback

Value	Termination cause
17	User Error
18	Host Request

CHƯƠNG 3 : MÔ HÌNH THỰC NGHIỆM VÀ KẾT LUẬN

3.1 Mô hình thực nghiệm:

3.1.1 Mô hình triển khai:



HÌNH 3.1 MÔ HÌNH THỰC NGHIỆM TRIỂN KHAI

Mô hình bao gồm :

- Máy Domain controler, làm luôn chức năng Radius SERVER**.**
  IP Address : 172.16.10.102
  Subnet Mask: 255.255.255.0

- VPN Sever, làm luôn chức năng Radius client
  Card mạng nội bộ :
  IpAddress: 172.16.10.103
  Subnet Mask: 255.255.255.0
  Card mạng internet:
  IpAddress: 10.0.0.103
  Subnet Mask: 255.255.255.0
- Máy Client (VPN Client)
  IpAddress: 10.0.0.0
  Subnet Mask: 255.255.255.0

3.1.2 Môi trường, các công cụ, phần mềm hỗ trợ:

- Môi trường triển khai Domain, sử dụng windown 2k3(Radius Server, Radius Client), Window XP (VPN Client). Domain name : nhom3.com

- VMWare: tiện ích hỗ trợ cài đặt, xây dựng máy ảo. Máy VMWare01(Radius Server) liên lạc với VMWare02 (Radius Client, VPN Server) thông qua card mạng 172.16.0.103 tượng trưng trong hệ thống nội bộ (cơ chế liên lạc 2 card mạng trong VMWare là NAT). VMWare02 (Radius Client ,VPN Server) liên lạc với VMWare03 (VPN Client) bằng card mạng 10.0.0.103 tượng trưng cho kết nối ngoài internet (cơ chế liên lạc 2 card mạng trong VMWare là Bridge).

- Phần mềm bắt gói tin WireShark: được cài đặt trên 3 máy để bắt các gói tin.

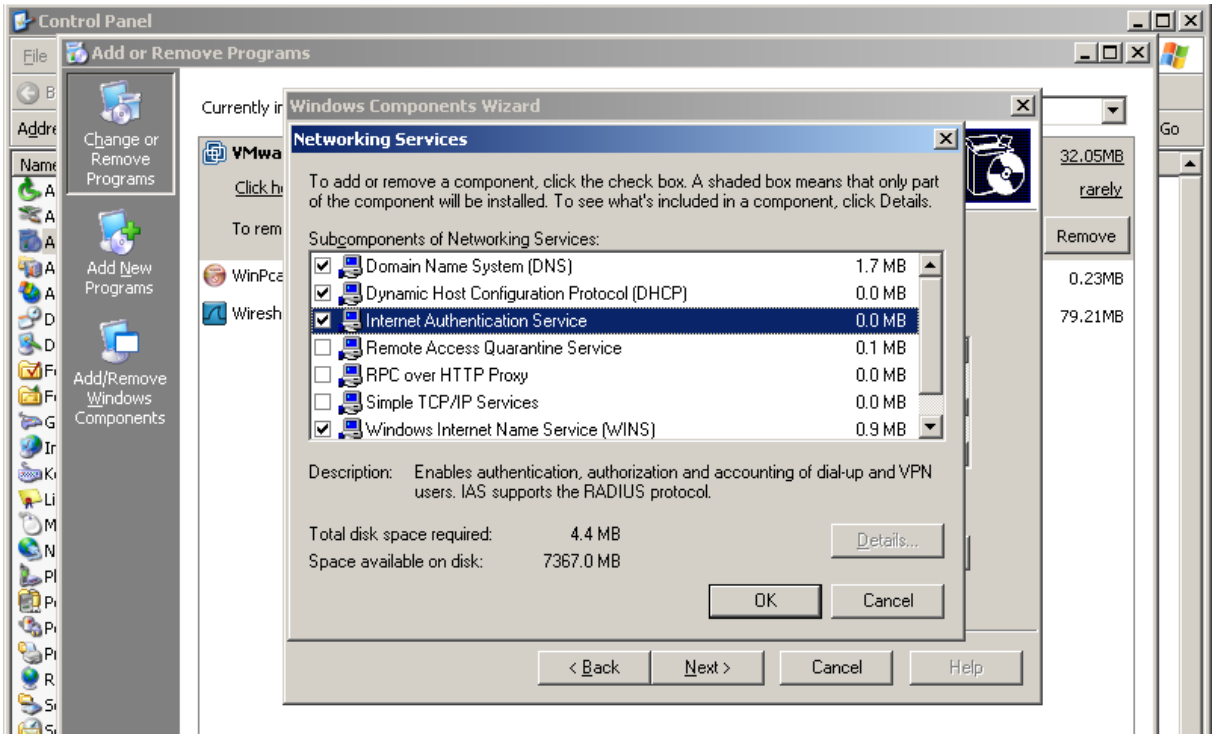
3.2 Các bước cài đặt, xây dựng mô hình:

3.2.1 Cài đặt DC làm chức năng Radius Server:

- Cài đặt máy VMWare01(win2k3) làm Domain Controller : ***Start → Run → gõ dcpromo***. Khai báo tên domain là ***nhom3.com***

- Trong quá trình nâng cấp lên dc, hệ thống tự cài đặt DNS Server.

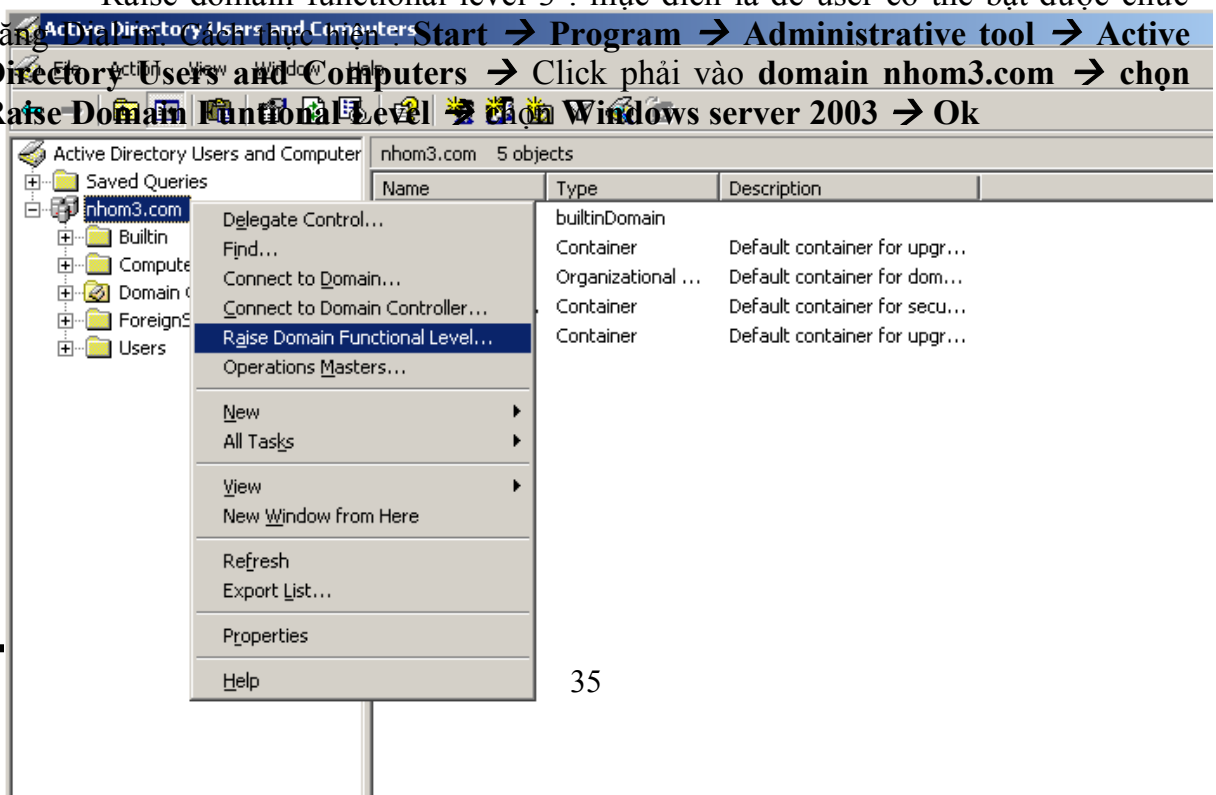
- Tiến hành cài đặt Radius Server thông qua cài đặt dịch vụ **Internet Authentication Service.**

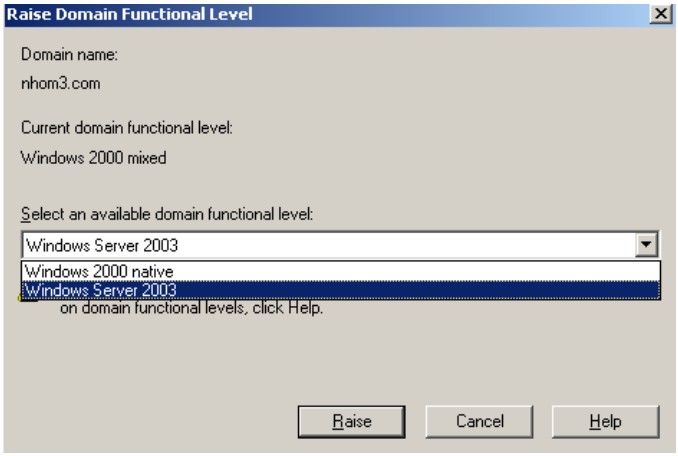


HÌNH 3.2 CHECK CHỌN INTERNET AUTHENTICATION SERVICE

Cách cài : **Start → Setting → Control panel → Add or remove programs → Add/remove Windows components → Networking services → Detail →** Check chọn **Internet authentication service → ok → next .** Hệ thống sẽ tiến hành tự động cài đặt Radius Server.

- Raise domain functional level 3 : mục đích là để user có thể bật được chức năng Dial-in. Cách thực hiện : **Start → Program → Administrative tool → Active Directory Users and Computers →** Click phải vào **domain nhom3.com → chọn Raise Domain Functional Level → chọn Windows server 2003 → Ok**





HÌNH 3.3 RAISE FUNCTIONAL LEVEL

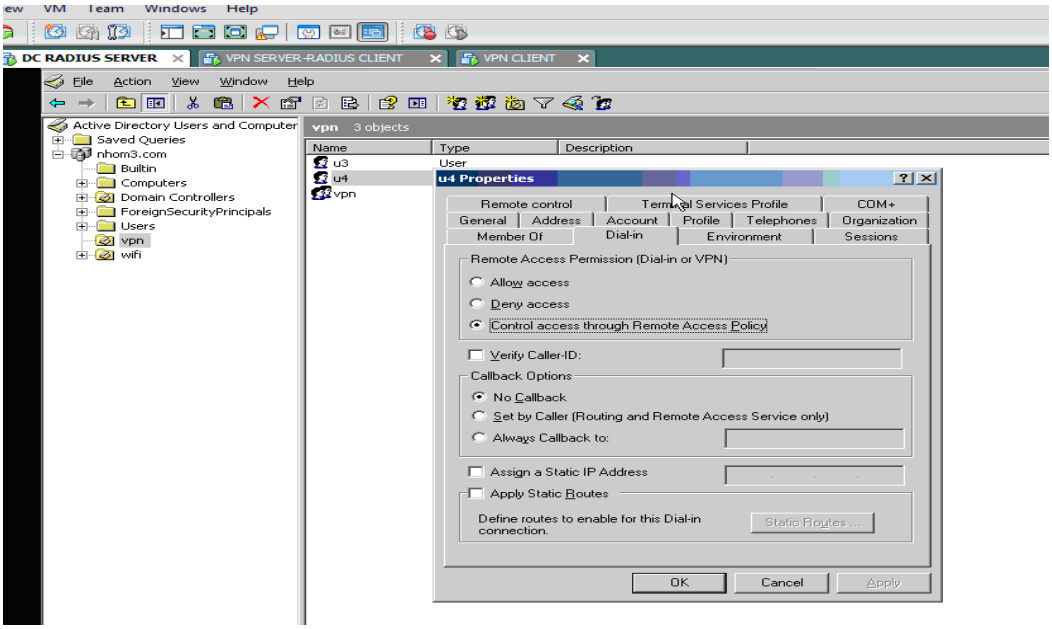
- Tạo OU,Group,users: Phục vụ cho quá trình kiểm tra bắt gói tin

Vào **Active Directory Users and Computers** → Bấm chọn vào **nhom3.com** chọn **new** → **Organizational Unit** → đặt tên OU là **vpn**

+ Click phải vào OU vpn chọn **new group** và đặt tên là **vpn**

+ Click phải vào OU vpn chọn **new user** và đặt tên là **u3** và **u4**

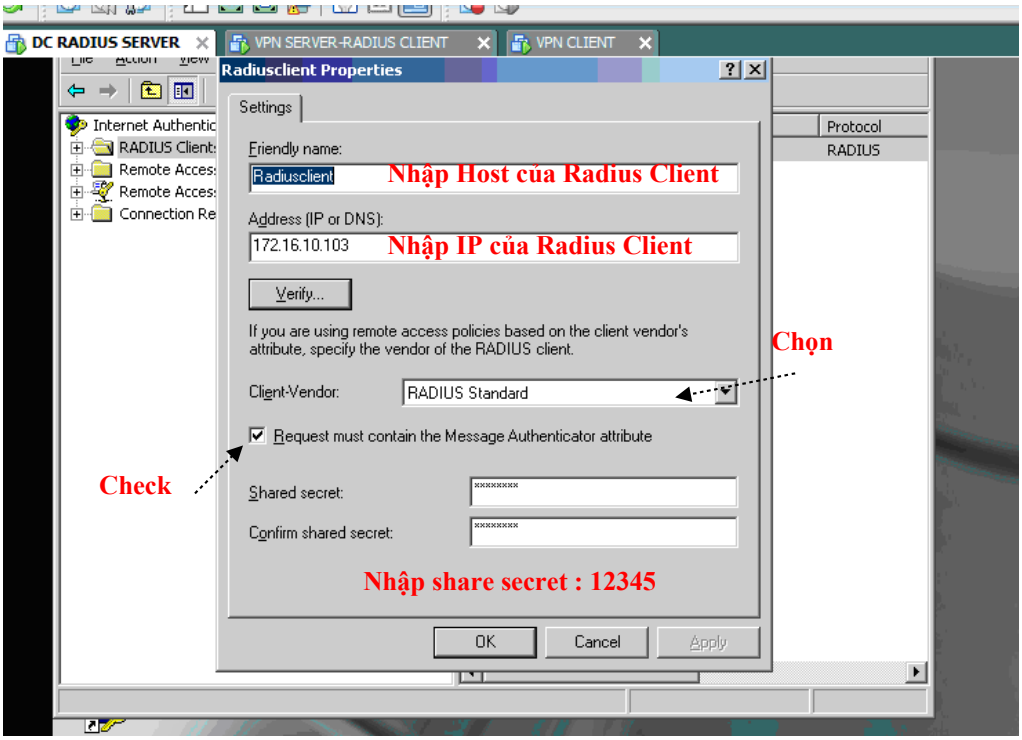
+ Click phải vào user **u3** và **u4** chọn **propeties** → chọn thẻ **Member Of** → **add** → chọn group **vpn** → **OK**,chọn thẻ **Dial-in** → check **Control access through Remote Access Policy** → **OK**



HÌNH 3.4 CẤU HÌNH USER CHO PHÉP SỬ DỤNG DỊCH VỤ REMOTE ACCESS

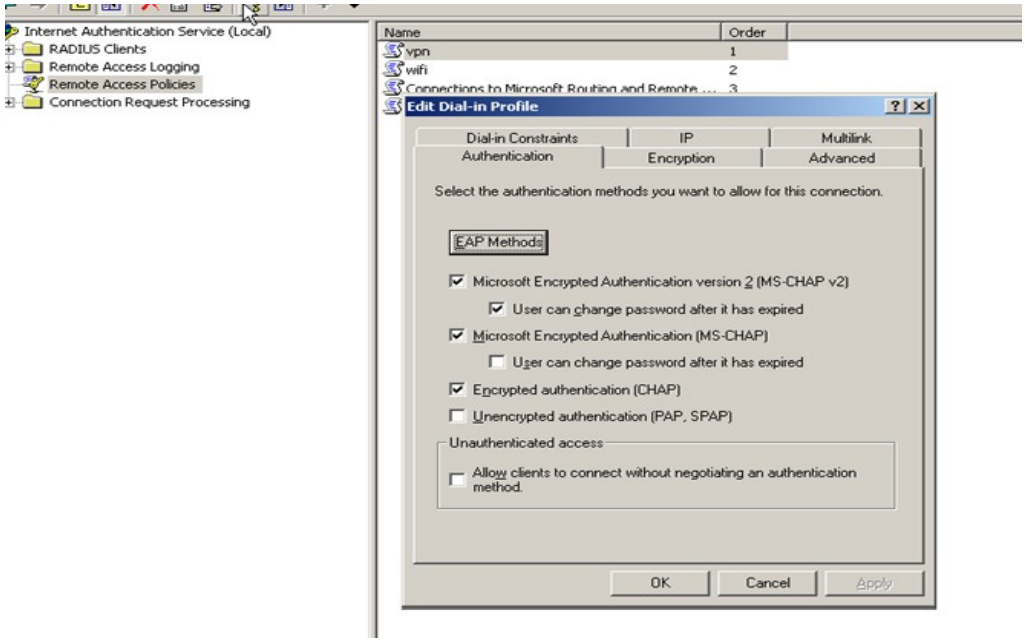
- Cấu hình Radius Server. **Start → Program → Administrative tool → Internet Authentication service →** Click phải **Internet Authentication service (Local)** chọn **Register Server in Active Directory → Ok**

+ Tạo mã share secret với Radius Client: Click phải **Radius client →** chọn **new Radius client.** Các thông số nhập theo hình 3.5.



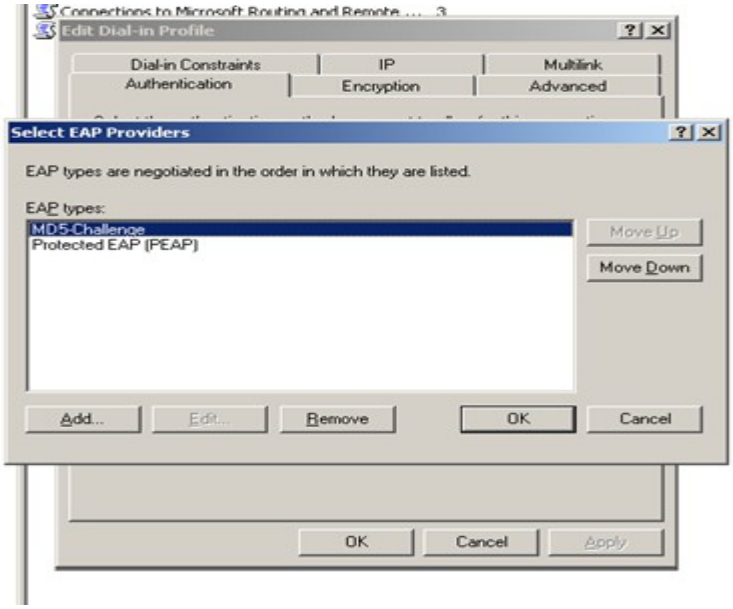
HÌNH 3.5 CẤU HÌNH SHARE SECRET GIỮA RADIUS SERVER/CLIENT

+ Chọn chế độ xác thực : Bấm vào **Remote access policy** → chọn **new remote access Policy** → Ra bảng **Remote access Policy Wizard** → Đặt tên **Policy named** là: **vpn** → chọn **VPN** → **group** → **add group** vpn vào → chọn **Next** → **ok.** Sau đó ta tiến hành chọn các phương thức chứng thực như bảng đưới đây trong hình 3.6.



HÌNH 3.6 CÁC TÙY CHỌN CƠ CHẾ CHỨNG THỰC

Để Add MD5 challenger và EAP: chọn EAP Method ở hình 3.6 .Chọn Add và OK



HÌNH 3.7 ADD TÙY CHỌN CHO CƠ CHẾ CHỨNG THỰC EAP

.

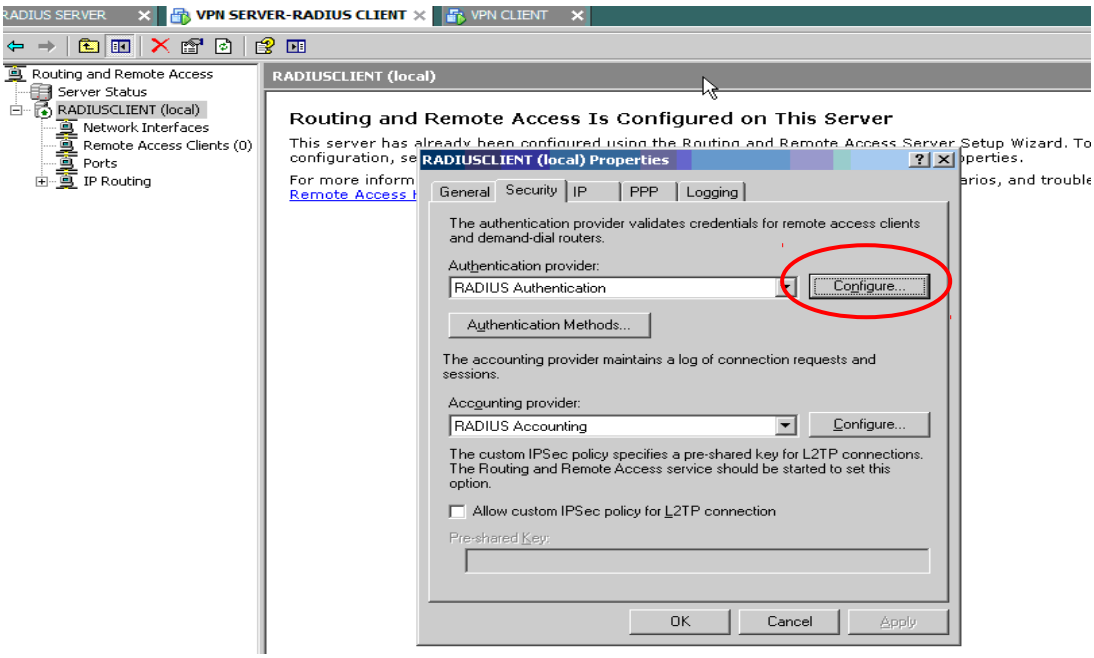
3.2.2 Cài đặt VPN Server làm chức năng Radius Server:

- Cho máy VMWare02 join vào domain **nhom3.com**

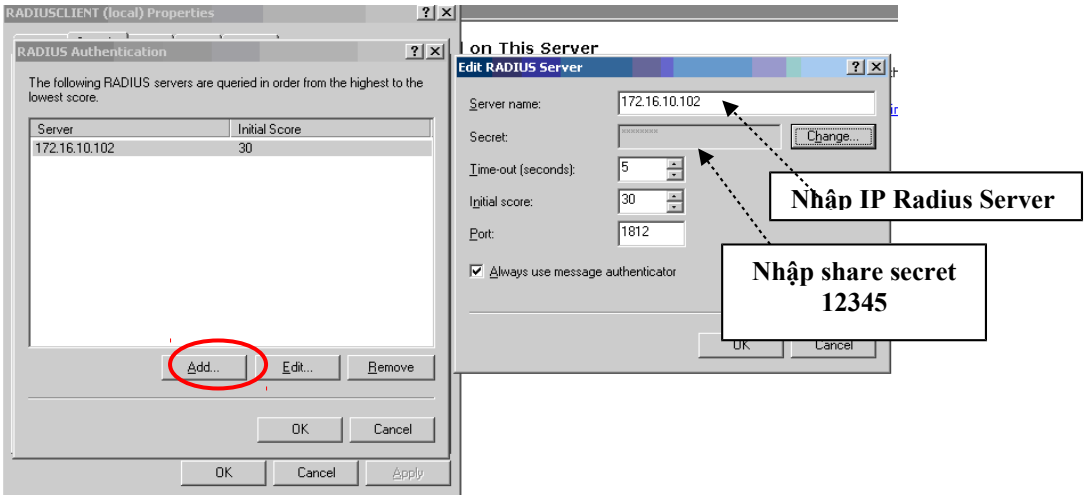
- Cài đặt làm Radius Client : **Start → Program → Administrative tool → Internet Authentication service →** Click phải **Internet Authentication service (Local)** chọn **Register Server in Active Directory → Ok**

- Cấu hình làm VPN Server : **Start → Program → Administrative tool → Routing And remote access →** Click phải chọn **New Routing And remote access →** chọn **custom config →** chọn **02 option VPN và Lan Routing → Start dv Routing And remote access**

Bấm phải vào **RADIUSCLIENT** chọn **propeties →** chọn thẻ **security** và cấu hình các thông số như hình 3.8 dưới đây.



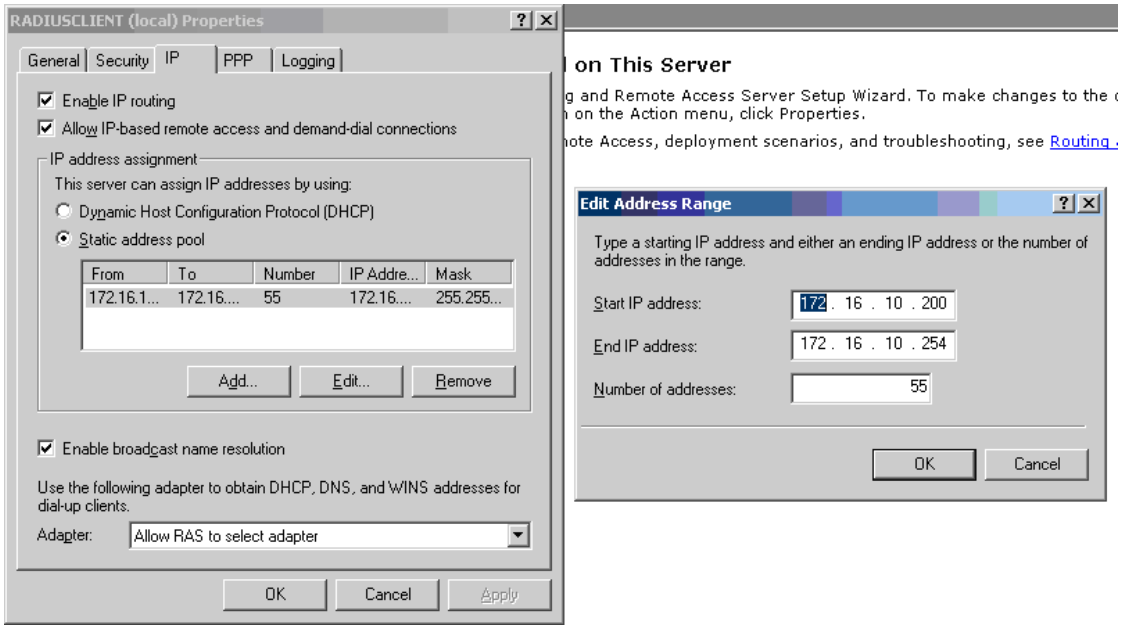
HÌNH 3.8 CẤU HÌNH RADIUS CLIENT



HÌNH 3.9 KHAI BÁO CÁC THÔNG SỐ CỦA SERVER

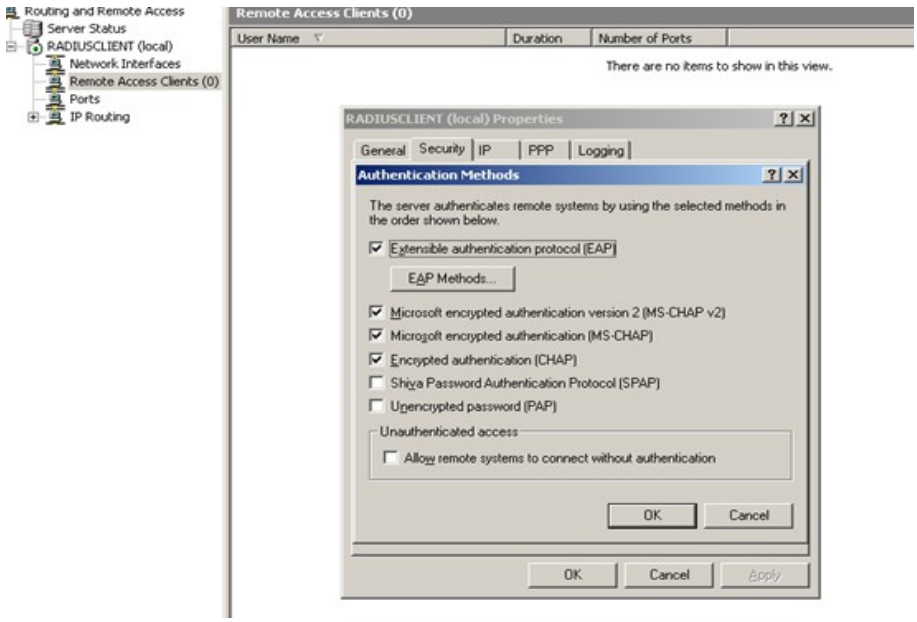
Ta chọn nút Configure ở hình 3.8 để tiên hành cấu hình các thông sô của Radius Server cho Radius Client. Ta chọn nút **Add** ở hình 3.9 và nhập IP Server Radius đồng thời nhập mã share secret đã khai báo trước đó khi cài đặt Server.

- Tại Tab IP ta chỉnh Pool IP để set IP cho các VPN Client kết nối vào như hình 3.10 dưới đây. Pool IP: 172.16.10.200-172.16.10.254. Sau đó restart dịch vụ Routing And Remote Access.



HÌNH 3.10 SET POOL IP CHO CÁC VPN CLIENT

- Cuối cùng tại thẻ Security ta tiến hành cấu hình các cơ chế hỗ trợ chứng thực tương như như ở Server.



HÌNH 3.11 CẤU HÌNH CƠ CHẾ HỖ TRỢ XÁC THỰC

* Tại VPN client ta thiết lập địa chỉ IP, thiết connection vpn, chuẩn bị quá trình bắt gói tin

HÌNH 3.12 GIAO DIỆN ĐĂNG NHẬP KẾT NỐI VPN TẠI REMOTE CLIENT

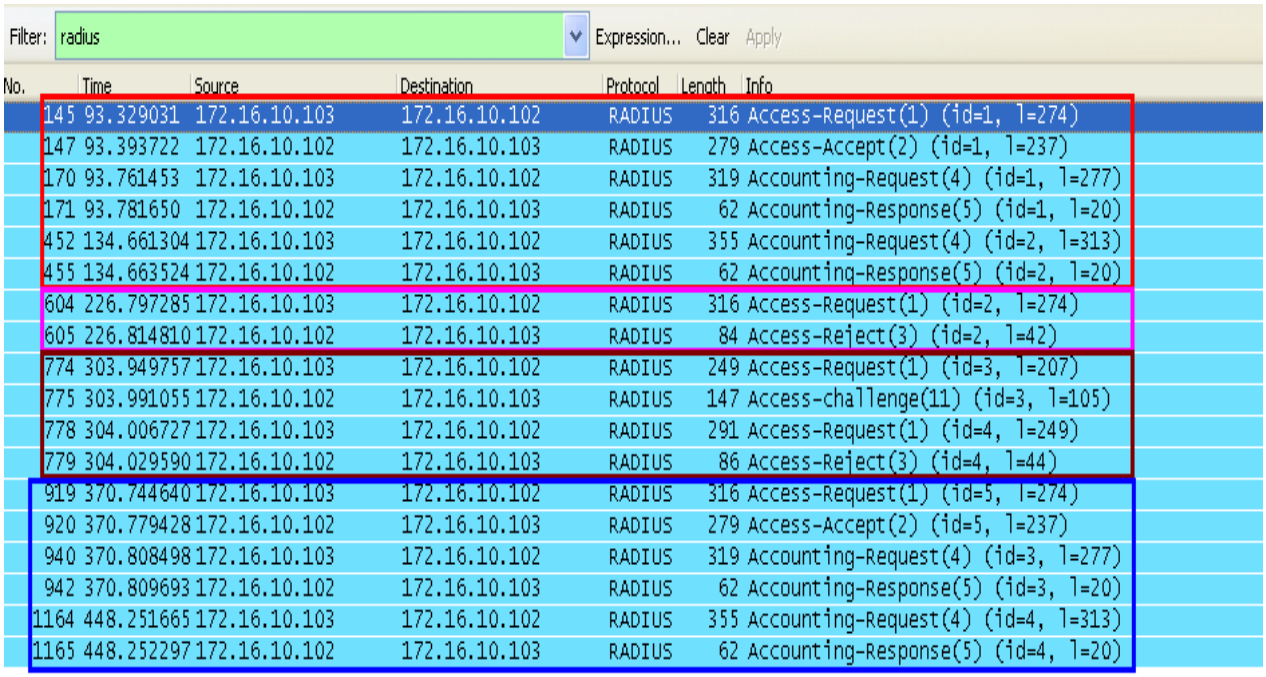
* Ngoài ra chương trinh bắt gói tin wire shark được cài đặt trên toàn bộ các máy để chuẩn bị qua bước bắt và phân tích gói tin tiếp theo.

3.3 Bắt và phân tích các gói tin:

3.3.1 Kịch bản bắt gói tin:

- User VPN nhập đúng Username/password
  -Access-Request
  -Access-Accept
  -Accounting-Request (bắt đầu dịch vụ)
  -Accounting-Response (chấp nhập bắt đầu từ server)
  -Accounting-Request(User yeu cau stop)
  -Accounting-Response(Server chap nhan stop)
- User VPN nhập Sai Username/password
  -Access-Request
  -Access-Reject
- Chỉnh thay đổi cơ chế m? hóa tại VPN Client (SD MD5-CHALLENGER,EAP) đúng Username/password
  -Access-Request
  -Access-Challenge
  -Access-Request
  -Access-Reject
- Kiểm tra lại nhập đúng Username/password

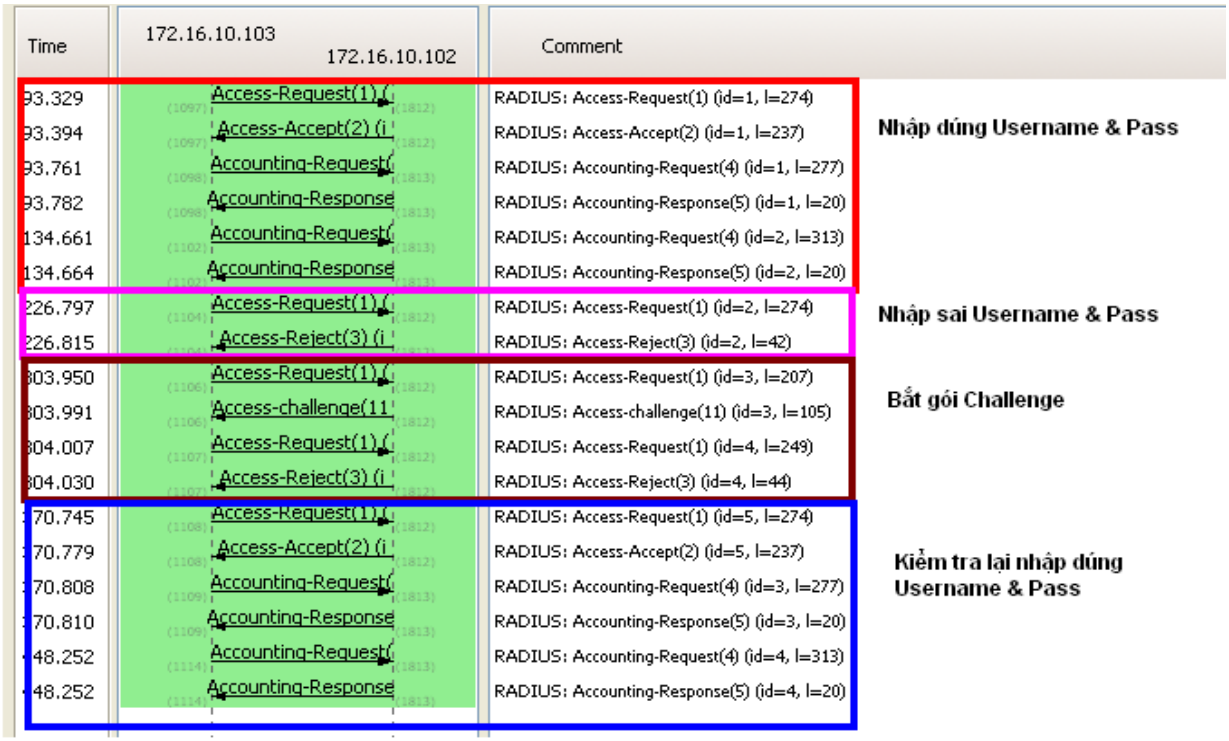
3.3.2 Kết quả bắt gói tin Radius:



Filter: radius — Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
145	93.329031	172.16.10.103	172.16.10.102	RADIUS	316	Access-Request(1) (id=1, l=274)
147	93.393722	172.16.10.102	172.16.10.103	RADIUS	279	Access-Accept(2) (id=1, l=237)
170	93.761453	172.16.10.103	172.16.10.102	RADIUS	319	Accounting-Request(4) (id=1, l=277)
171	93.781650	172.16.10.102	172.16.10.103	RADIUS	62	Accounting-Response(5) (id=1, l=20)
452	134.661304	172.16.10.103	172.16.10.102	RADIUS	355	Accounting-Request(4) (id=2, l=313)
455	134.663524	172.16.10.102	172.16.10.103	RADIUS	62	Accounting-Response(5) (id=2, l=20)
604	226.797285	172.16.10.103	172.16.10.102	RADIUS	316	Access-Request(1) (id=2, l=274)
605	226.814810	172.16.10.102	172.16.10.103	RADIUS	84	Access-Reject(3) (id=2, l=42)
774	303.949757	172.16.10.103	172.16.10.102	RADIUS	249	Access-Request(1) (id=3, l=207)
775	303.991055	172.16.10.102	172.16.10.103	RADIUS	147	Access-challenge(11) (id=3, l=105)
778	304.006727	172.16.10.103	172.16.10.102	RADIUS	291	Access-Request(1) (id=4, l=249)
779	304.029590	172.16.10.102	172.16.10.103	RADIUS	86	Access-Reject(3) (id=4, l=44)
919	370.744640	172.16.10.103	172.16.10.102	RADIUS	316	Access-Request(1) (id=5, l=274)
920	370.779428	172.16.10.102	172.16.10.103	RADIUS	279	Access-Accept(2) (id=5, l=237)
940	370.808498	172.16.10.103	172.16.10.102	RADIUS	319	Accounting-Request(4) (id=3, l=277)
942	370.809693	172.16.10.102	172.16.10.103	RADIUS	62	Accounting-Response(5) (id=3, l=20)
1164	448.251665	172.16.10.103	172.16.10.102	RADIUS	355	Accounting-Request(4) (id=4, l=313)
1165	448.252297	172.16.10.102	172.16.10.103	RADIUS	62	Accounting-Response(5) (id=4, l=20)

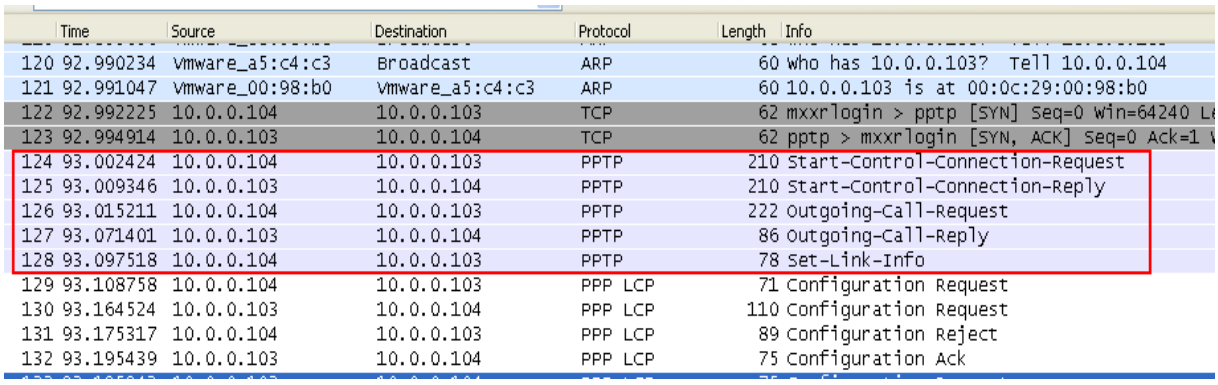
HÌNH 3.13 KẾT QUẢ BẮT CÁC GÓI TIN RADIUS



HÌNH 3.14 SƠ ĐỒ LUỒNG DỮ LIỆU CÁC GÓI TIN RADIUS

3.3.3 Phân tích gói tin:

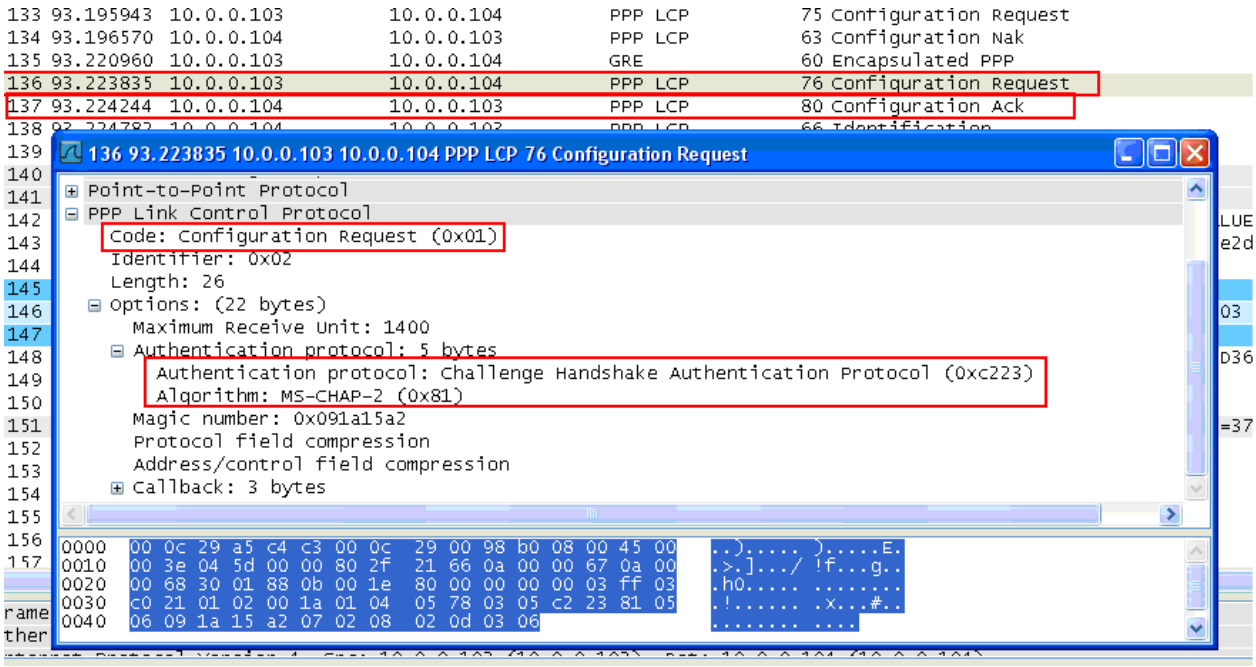
Do mô hình sử dụng VPN nên trước khi quá trình gửi username password lên hệ thống là quá trình thiết lập kết nối thông qua giao thức PPTP, (một trong những giao thức VPN sử dụng trong quá trình hoạt động)



	Time	Source	Destination	Protocol	Length	Info
120	92.990234	Vmware_a5:c4:c3	Broadcast	ARP	60	Who has 10.0.0.103? Tell 10.0.0.104
121	92.991047	Vmware_00:98:b0	Vmware_a5:c4:c3	ARP	60	10.0.0.103 is at 00:0c:29:00:98:b0
122	92.992225	10.0.0.104	10.0.0.103	TCP	62	mxxrlogin > pptp [SYN] Seq=0 Win=64240 L
123	92.994914	10.0.0.103	10.0.0.104	TCP	62	pptp > mxxrlogin [SYN, ACK] Seq=0 Ack=1
124	93.002424	10.0.0.104	10.0.0.103	PPTP	210	Start-Control-Connection-Request
125	93.009346	10.0.0.103	10.0.0.104	PPTP	210	Start-Control-Connection-Reply
126	93.015211	10.0.0.104	10.0.0.103	PPTP	222	Outgoing-Call-Request
127	93.071401	10.0.0.103	10.0.0.104	PPTP	86	Outgoing-Call-Reply
128	93.097518	10.0.0.104	10.0.0.103	PPTP	78	Set-Link-Info
129	93.108758	10.0.0.104	10.0.0.103	PPP LCP	71	Configuration Request
130	93.164524	10.0.0.103	10.0.0.104	PPP LCP	110	Configuration Request
131	93.175317	10.0.0.104	10.0.0.103	PPP LCP	89	Configuration Reject
132	93.195439	10.0.0.103	10.0.0.104	PPP LCP	75	Configuration Ack

HÌNH 3.15 GIAO THỨC PPTP THIẾT LẬP KẾT NỐI

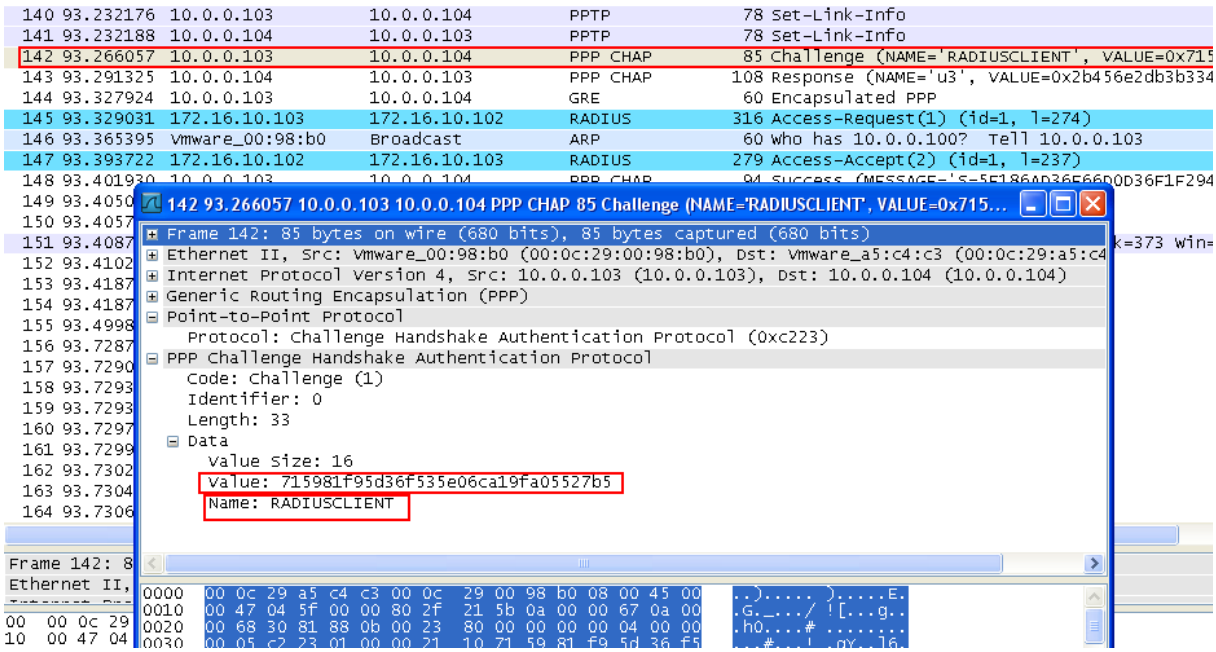
Sau đó giữa Remote user và Radius client tiến hành trao đổi thông tin để quyết định sử dụng giao thức xác thực nào



HÌNH 3.16 GỬI YÊU CẦU VỀ PHƯƠNG THỨC XÁC THỰC

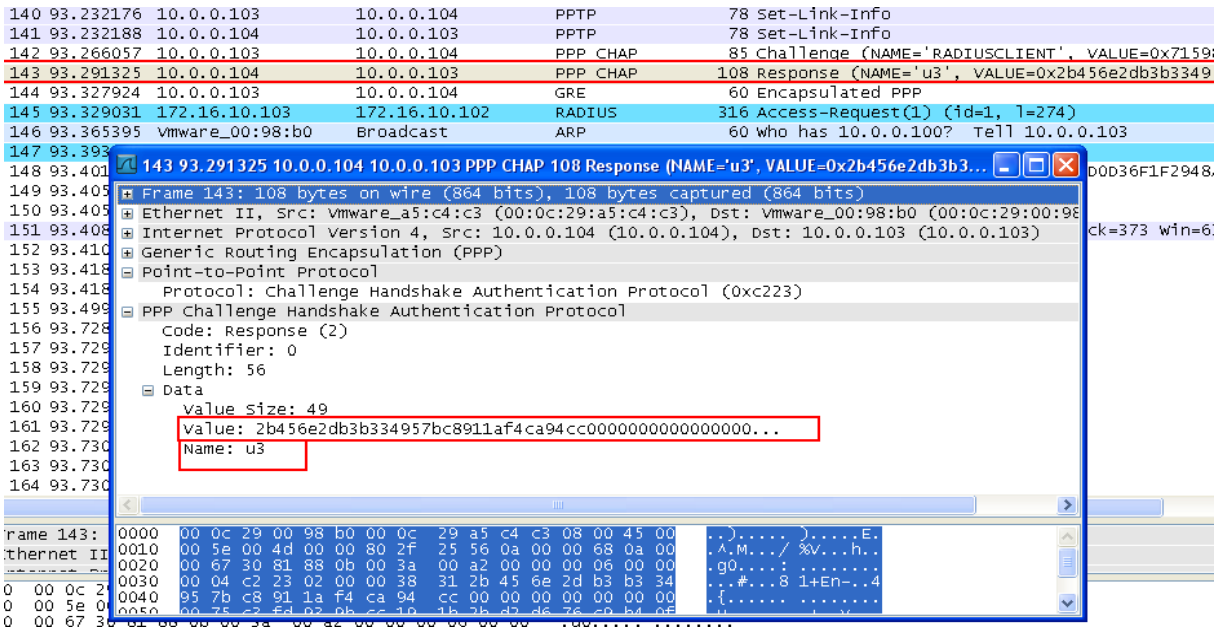
Theo như hình 3.16 Radius Client (10.0.0.103) gửi yêu cầu Configuration Request đến Client (10.0.0.104) yêu cầu sử dụng phương thức xác thực đó là MS-CHAPv2. Và bước tiếp theo Client sẽ phản hồi thông điệp Configuration Ack để chấp nhận yêu cầu. Đến đây hoàn thành bước bắt tay xác định phương thức xác thực.

Tiếp theo đó Radius Client sẽ thực hiện gửi một PPP CHAP Challenge có chứa một giá trị ngẫu nhiên như hình 3.17.



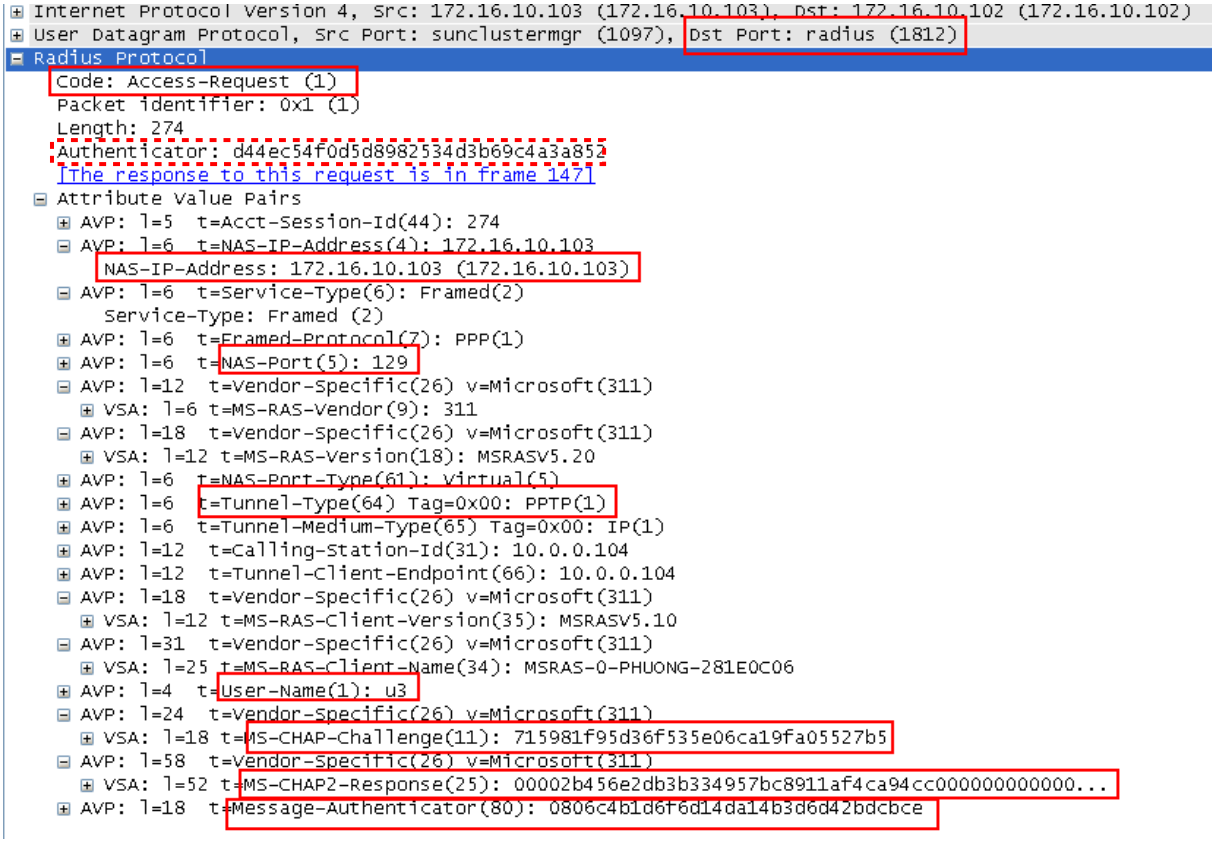
HÌNH 3.17 RADIUS CLIENT GỬI MỘT GIÁ TRỊ NGẪU NHIÊN CHO CLIENT

Tại máy người dùng sau khi nhận được thông điệp có chứa giá trị challenge này tiến hành phản hồi username và password lên Radius client với trường giá trị Name : u3 là user name của người dùng và giá trị value đươc tính bằng Hash MD5 (password người dùng + giá trị challenge nhận được).



HÌNH 3.18 NGƯỜI DÙNG GỬI USERNAME VÀ PASSWORD LÊN RADIUS CLIENT

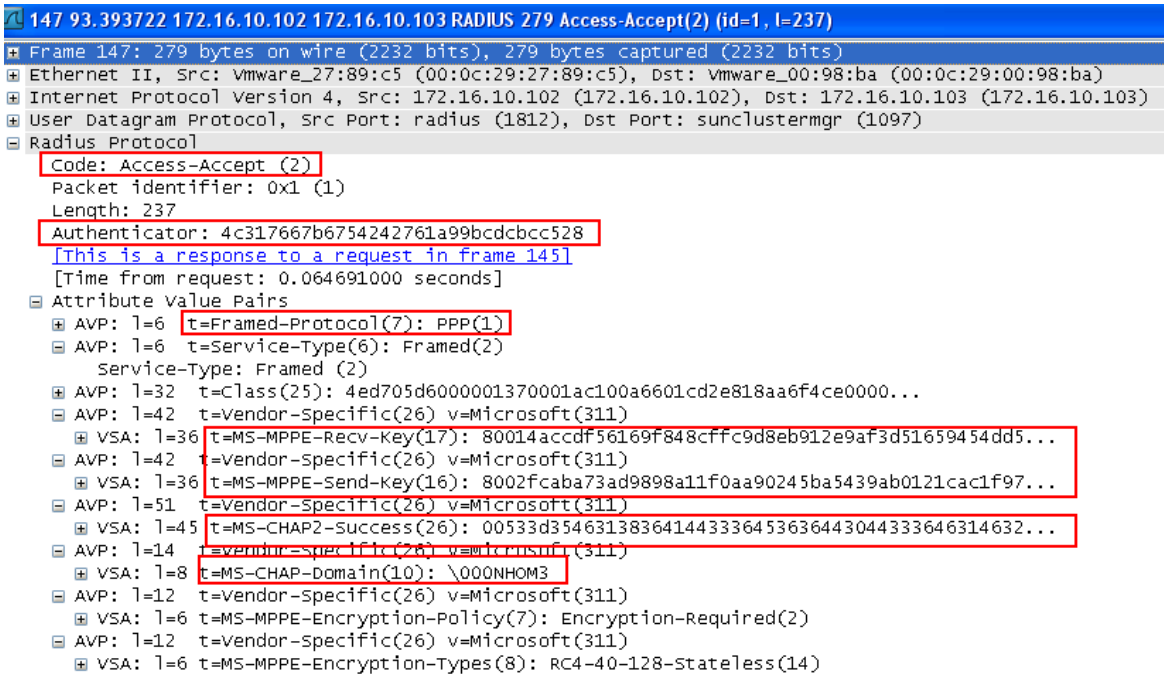
Khi Radius Client nhận được thông tin của người dùng gửi lên hệ thống sẽ gửi Access Request lên cho Radius Server để xin xác thực cho người dùng.



HÌNH 3.19 RADIUS CLIENT GỬI YÊU CẦU ACCES-REQUEST CHO RADIUS SERVER

Radius sẽ lấy thông tin username là u3 gán vào giá trị trường **User-name(1)**, giá trị challenge khởi tạo ban đầu gửi cho user lưu vào trường **MS-CHAP-Challenge(11).** Giá trị phản hồi (hàm băm MD5 của password người dùng và giá trị challage ban đầu) lưu vào trường **MS-CHAP2-Response**. Ngoài ra còn một số trường như **NAS-IP-Address** – lưu địa chỉ IP của Radius Client và **NAS-Port** là port mà user đang kết nối đến trên Radius Client. Và trường giá trị Authenticator, ở đây gọi là **Request Authenticator** được Radius Client tạo ngẫu nhiên. Ngoài ra ta còn thấy một trường giá trị **Message-Authenticator(80)** trường này phục vụ cho việc xác nhận yêu cầu này là do chính Radius Client gửi đến. Cách tính Message-Authenticator = HMAC-MD5 (Type + Identifier + Length + Request Authenticator + Attributes).

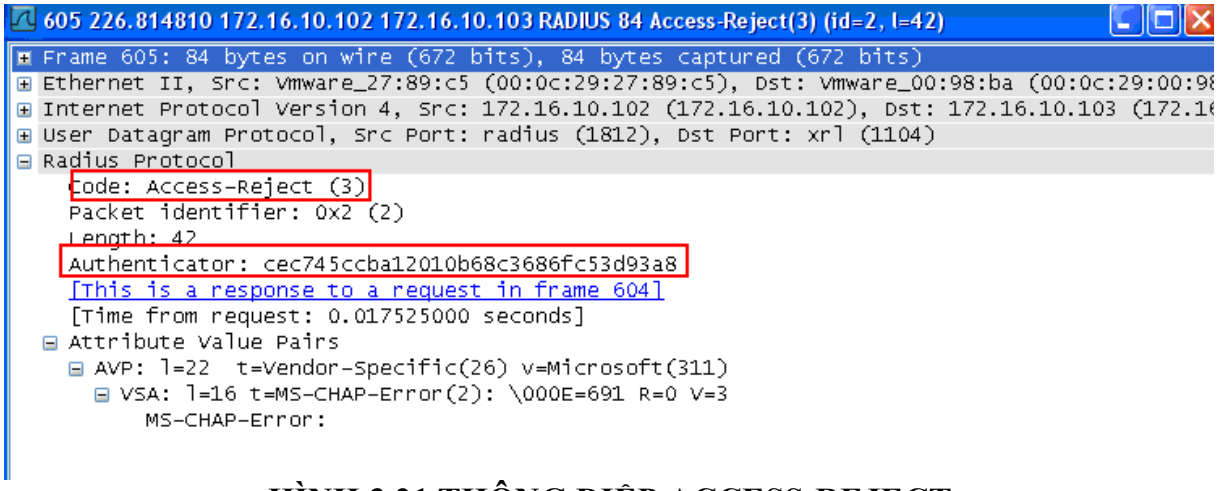
Sau khi nhận được yêu cầu Access Request gửi đến Radius Server tiến hành một loạt các hành động nhằm kiểm tra thông tin xác thực từ Radius Client gửi đến, so sánh thông tin với CSDL, cũng như kiểm tra một số các tùy chọn. Nếu tất cả hợp lệ Radius sẽ gửi thông điệp Access-Accept để chấp nhận xác thực người dùng và cho phép người dùng truy cập hệ thống.



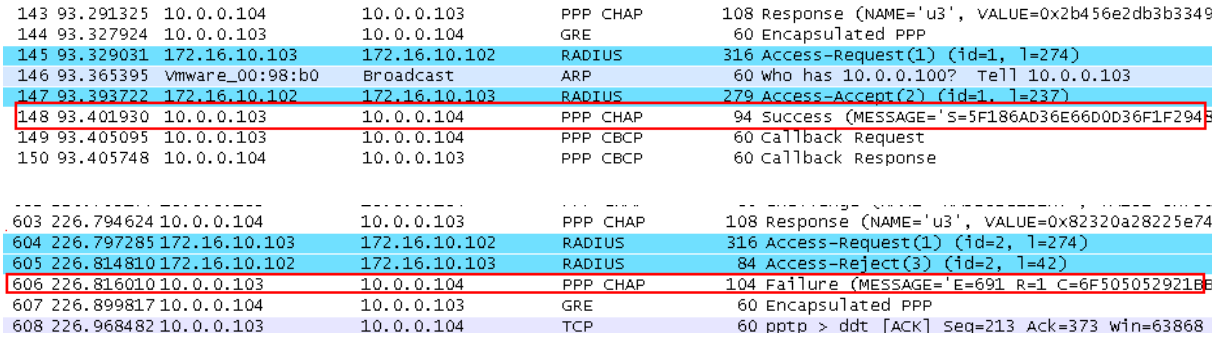
HÌNH 3.20 RADIUS SERVER PHẢN HỒI ACCESS ACCEPT

Trong phản hồi Access-Accept trường Authenticator (Response Authenticator) đã thay đổi giá trị do đã được tính lại **MD5(Code + Identifier + Length + RequestAuthenticator + Attributes + Secret(share secret).** Với Secret là '12345' đã tạo khi cài đặt hệ thống. Và thông điệp Access-Accept sẽ thường chứa các trường giá trị thông tin cấu hình dịch vụ mà người dùng yêu cầu. **MS-CHAP2-Sucess(26)** là mã phản hồi thành công do Server gửi. Radius Client tiếp tục forward mã này về người dùng. Bên cạnh đó ta thấy 2 trường **MS-PPE-Recv-Key(17)** và **MS-PPE-Send-Key(16)**, do trong phần cấu hình ta check có hỗ trợ PEAP nên 2 trường này phục vụ cho cơ chế EAP.

Đối với những trường hợp không hợp lệ Server sẽ phản hồi thông điệp Access-Reject để từ chối người dùng. Thông điệp này ko chứa bất cứ trường thuộc tính nào.



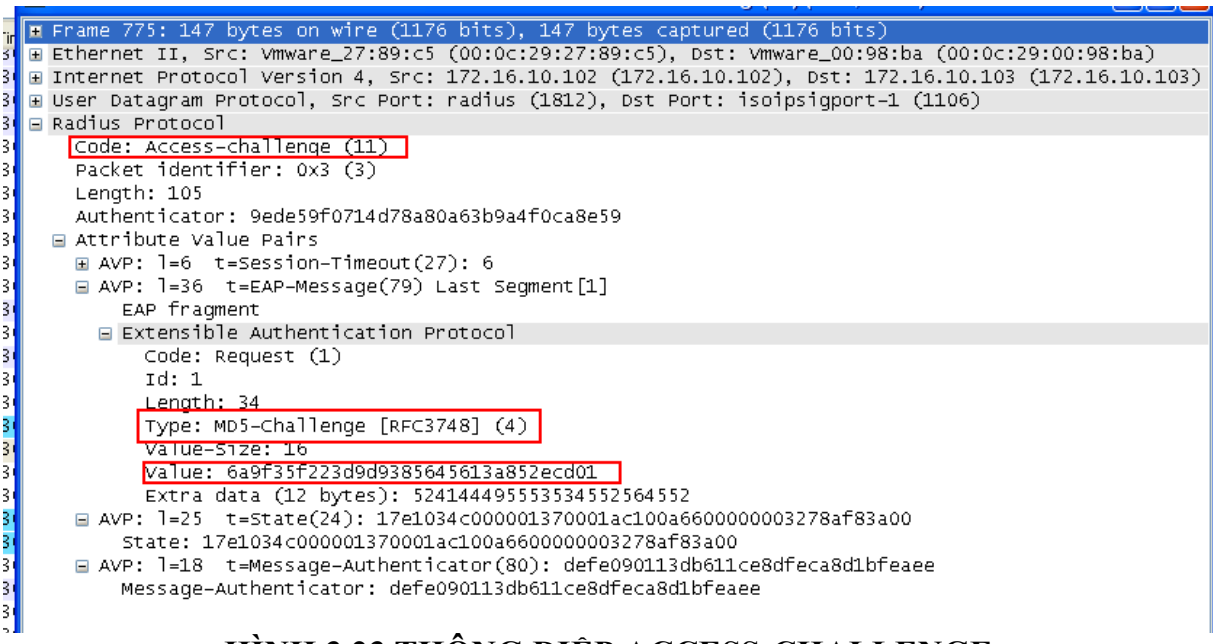
HÌNH 3.21 THÔNG ĐIỆP ACCESS-REJECT



HÌNH 3.22 RADIUS PHẢN HỒI CHO NGƯỜI DÙNG KẾT QUẢ XÁC THỰC

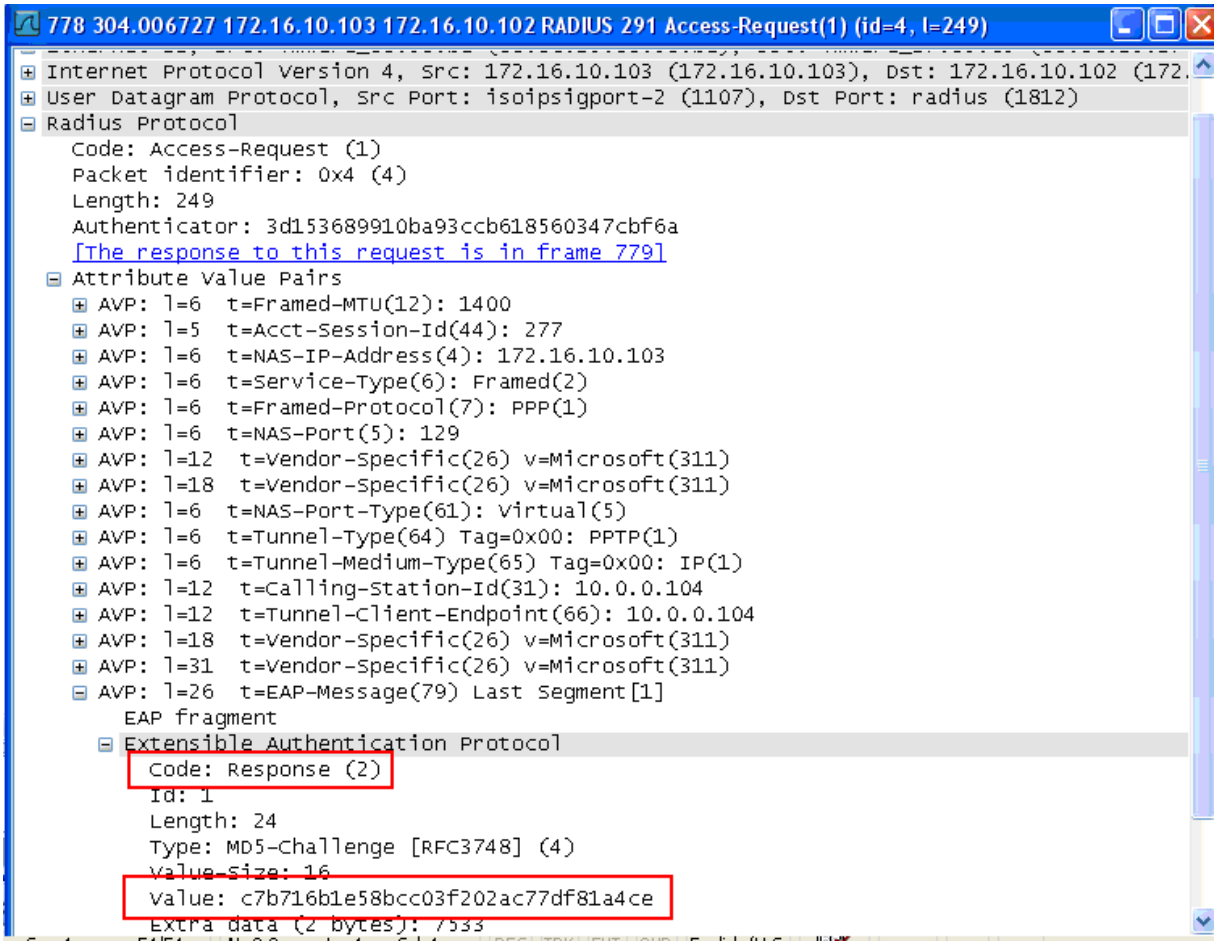
Dù nhận được phản hồi Access-Accept hay một phản hồi Access-Reject thì Radius Client đều phản hồi ngược lại kết quả cho người dùng (hình 3.22).

Trong một số trường hợp để đảm bảo tính xác thực Radius Server sẽ gửi một yêu cầu Challenge yêu cầu người dùng phản hồi và dựa vào kết quả phản hồi sẽ quyết định bước tiếp theo.



HÌNH 3.23 THÔNG ĐIỆP ACCESS-CHALLENGE

Trong thông điệp cho biết thuật toán sư dụng và một chuỗi giá trị ngẫu nhiên. Phần mềm hay thiết bị của người dùng hợp lệ sẽ dựa vào giao thức này và giá trị challenge này sẽ tính toán và gửi kết quả lên Radius Client. Radius Client chuyển tiếp kết quả tính toán lên Radius Server. Tại Radius Server, quá trình tự tính toán kết quả diễn ra, sau đó so sánh với kết quả nhận được để quyết định tiếp bước tiếp theo (Access Accept khi hợp lệ, Access-Reject để từ chối, hoặc một Access Challenge khác)

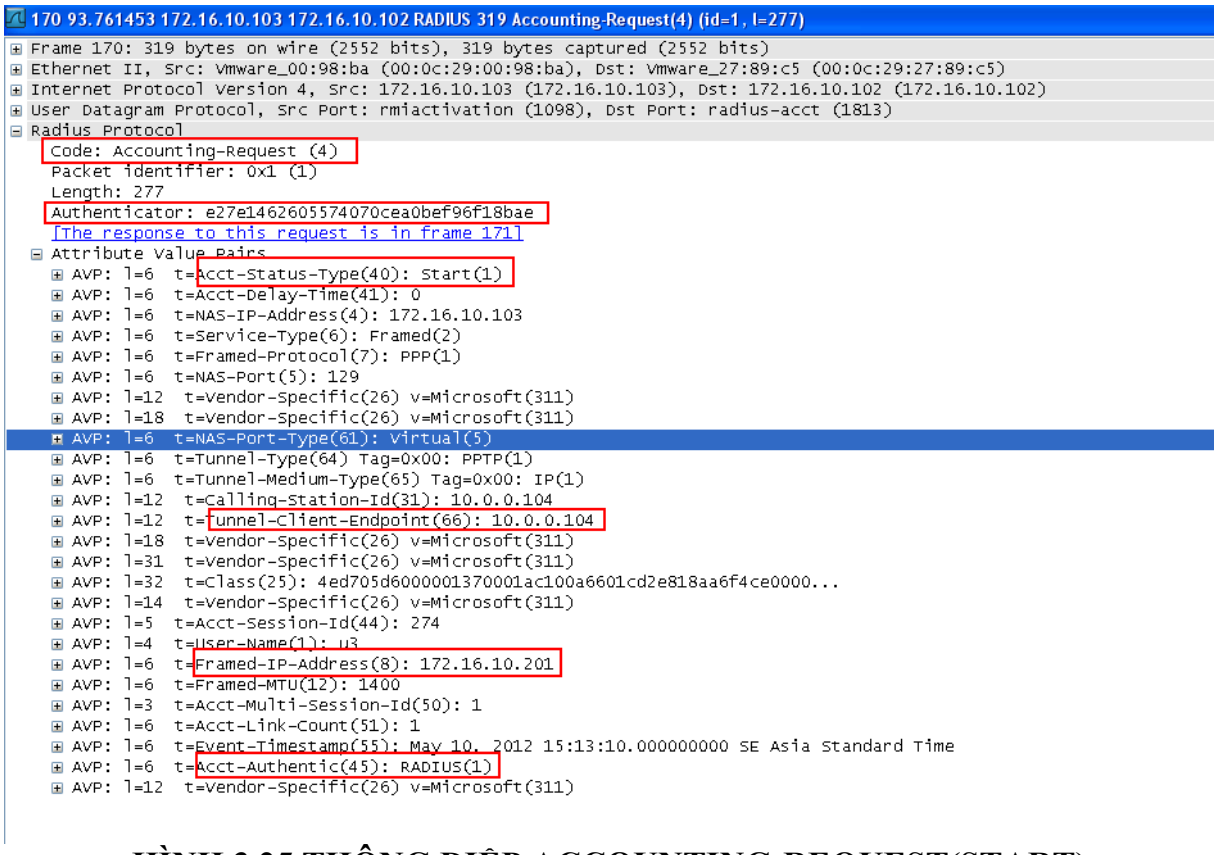


HÌNH 3.24 RADIUS GỬI LẠI ACCESS REQUEST ĐỂ PHẢN HỒI KẾT QUẢ LÊN RADIUS SERVER

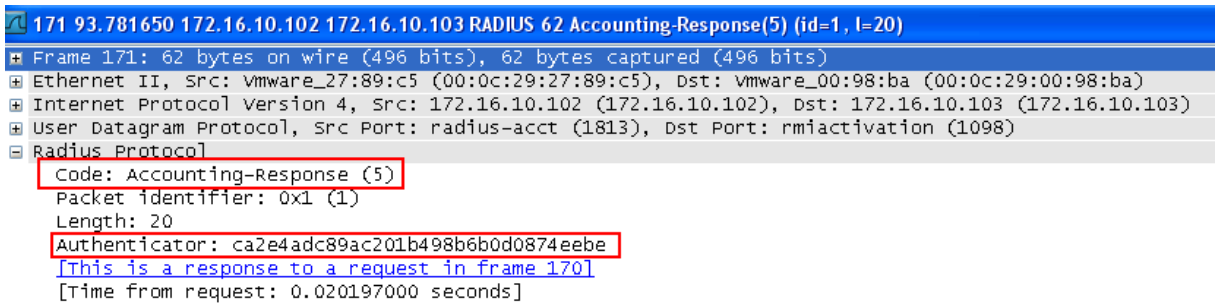
Sau khi hoàn thành quá trình xác thực - cấp quyền truy cập cho người dùng, nếu được cấu hình chức năng accounting, radius sẽ gửi thông điệp Accounting Request yêu cầu Radius Server bắt đầu dịch vụ này.

Trường thuộc tính **Acct-status-type(40)** là **start**, nghĩa là Radius client yêu cầu Radius bắt đầu thực hiện chức năng Accounting cho người dùng. **Tunnel-Client-Endpoint(66)** : trường này chỉ địa chỉ thực của remote client ngoài internet đang kết nối đến hệ thống. **Frame-Ip-Address(8)** : trường này chỉ địa chỉ IP của VPN server phát cho VPN Client. Trường Authenticator (Request Authenticatior) được tính bằng **MD5 (Code + Identifier + Length + 16 zero octets + Attributes + Secret).** (Hình 3.25)

Khi nhận được yêu cầu gửi đến, Server Radius sẽ phản hồi thông điệp Accouting-Response để thừa nhận đã nhận yêu cầu trước đó do Radius Client gửi. Và trong Accounting không chứa bất cứ trường thuộc tính nào. Giá trị Authenticator (Response Authenticator) phục vụ cho việc xác nhận 2 bên tham gia và chống chối bỏ, có cơ chế như một chữ kí điện tử. Được tính bằng **MD5(Code + Identifier + Length + RequestAuthenticator + Attributes + Secret).**



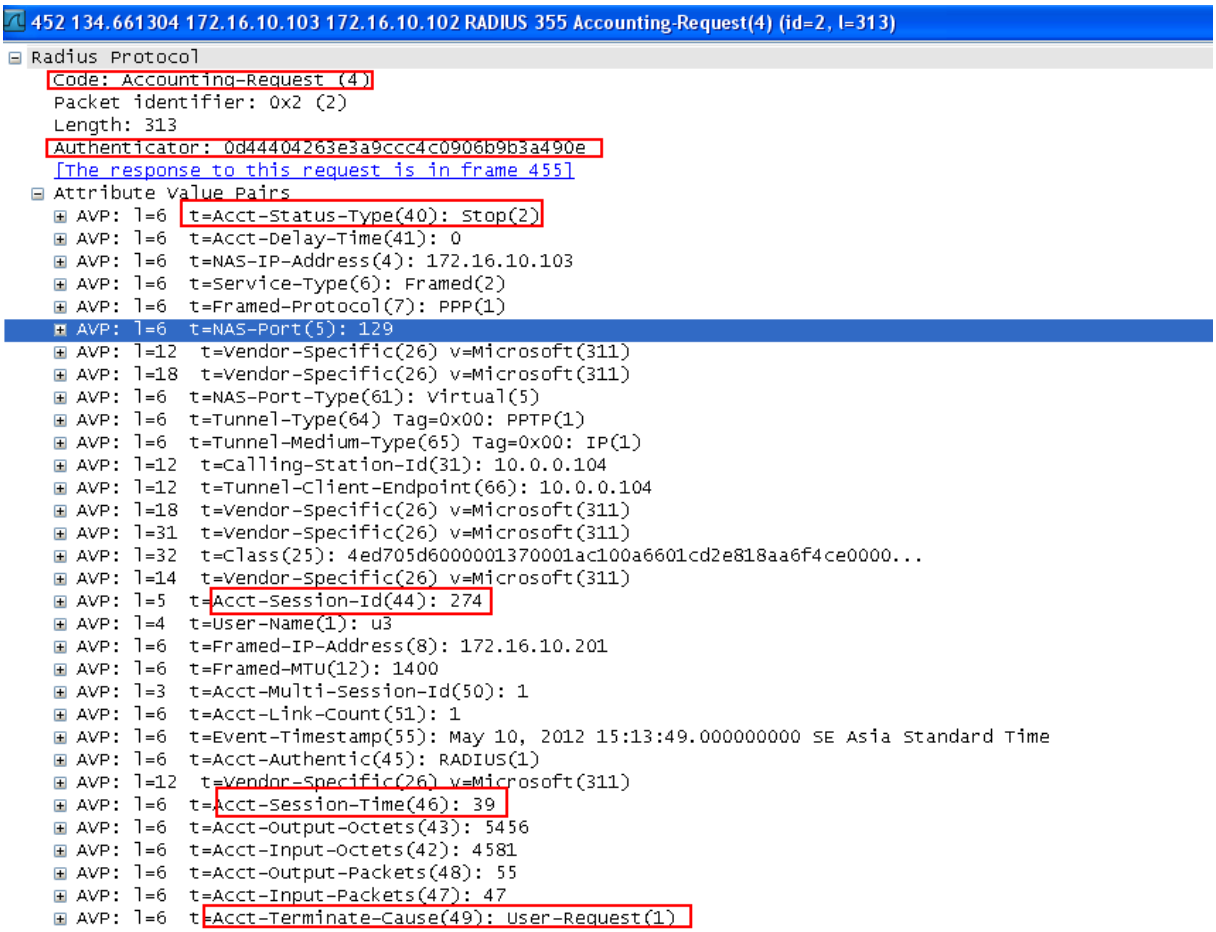
HÌNH 3.25 THÔNG ĐIỆP ACCOUNTING-REQUEST(START)



HÌNH 3.26 THÔNG ĐIỆP PHẢN HỒI ACCOUNTING-RESPONSE

Khi user ngắt kết nối VPN, Radius gửi đến Radius Server một Accouting-Request với thông số thuộc tính **Acct-status-type(40)** là **stop** để yêu cầu server dừng dịch vụ. Đồng thời gửi kèm các thuộc tính về phiên làm việc. **Acct-Session-ID (44)**: mã id để phân biệt giữa các phiên làm việc giữa các người dùng khác nhau. **Acc-Terminate-Cause(49)** : chỉ nguyên nhân ngừng cung cấp dịch vụ, ở đây có giá trị là User-Request tức là chính người dùng yêu cầu tạm ngừng dịch vụ.

Nếu Radius nhận được thông điệp yêu cầu ngừng cung cấp dịch vụ, đồng thời lưu trữ thành công các thông tin trạng thái về phiên làm việc sẽ gửi một phản hồi Accouting-Response thừa nhận đã nhận yêu cầu trước đó và kết thúc quá trình kết nối với người dùng.



HÌNH 3.27 RADIUS CLIENT GỬI ACCOUTING REQUEST (STOP)

3.4 Kết luận

- Mô hình luồng dữ liệu trong mô hình thực nghiệm hoàn toàn phù hợp với mô hình lý thuyết ở chương 2. Thứ tự các gói tin bắt được, các thuộc tính trong các gói tin đều mô tả đúng như phần lý thuyết.

- Việc xác thực, mã hóa, chống chối bỏ và toàn vẹn dữ liệu của Radius phụ thuộc chủ yếu vào trường Authenticator. Auth Request phục vụ để mã hóa password cho user. Còn trong Auth Response trường này dùng vào việc chống chối bỏ giữa các bên tham gia cũng như kết hợp với key secret (share key) đảm bảo cho tính toàn vẹn dữ liệu trong quá trình gửi gói tin.

- Tuy hỗ trợ đa dạng về yêu cầu người dùng cũng như nhiều thiết kế ứng dụng mạng nhưng chỉ phù hợp cho những mô hình hệ thống lớn.

- Đã và đang là giao thức tiêu chuẩn trong việc xác thực người dùng truy cập từ xa nhưng với bản chất là một giao thức mở rộng, chúng ta cùng chờ những mở rộng, những cải tiến để hạn chế bớt các khuyết điểm hạn còn tồn đọng, hoặc biết đâu một giao thức mới có thể thay thế hoàn toàn giao thức Radius trong tương lai sắp đến.

TÀI LIỆU THAM KHẢO

- **TÀI LIỆU RFC 2865, 2866**

- **http://en.wikipedia.org/wiki/RADIUS**

- **"The Beginnings and History of RADIUS" Interlink Networks. Retrieved 2009-04-15.**

- ***RADIUS - Securing Public Access to Private Resources*** **(O'Reilly & Associates)**