Chưa hoàn thiện (30%), còn sai sót

Bài 1: Bài mở đầu

1. Khái niệm cơ bản
   1. An toàn mạng máy tính

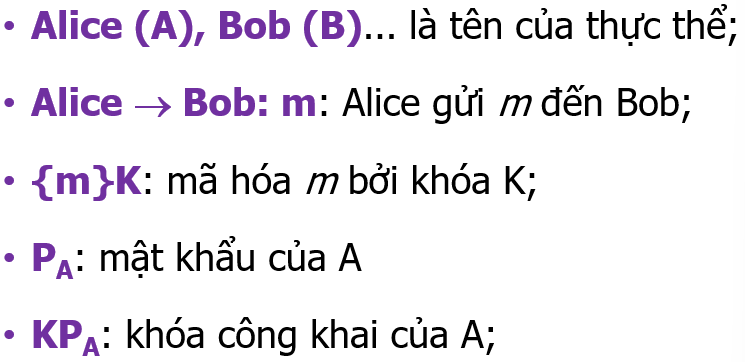
* Là việc ngăn chặn và giám sát các truy cập trái phép, sự lạm dụng, sửa đổi hoặc làm gián đoạn hoạt động của mạng máy tính và các tài nguyên mạng
* Được đảm bảo bằng 1 tập tập hợp các chính sách và giải pháp kỹ thuật
  1. Tấn công
* Là hành động có chủ ý nhằm phá vỡ sự an toàn của thông tin, hệ thống thông tin được bảo vệ
* Tấn công bị động:
  + Là kiểu tấn công chặn bắt thông tin như nghe trộm và quan sát truyền tin
  + Mục đích của kẻ tấn công là biết được thông tin truyền qua mạng
  + Tấn công bị động rất khó bị phát hiện bởi vì chúng không thay đổi bất kì dữ liệu nào
* Tấn công chủ động:
  + Là tấn công nhằm ngăn chặn, thay đổi các thông tin và chén các thông tin giả mạo
  + Tấn công chủ động dễ phát hiện nhưng lại khó ngăn chặn tuyệt đối
  1. Dịch vụ an toàn
* Là dịch vụ nâng cao an toàn của các hệ thống. mỗi dịch vụ an toàn sử dụng 1 hay nhiều kĩ thuật đảm bảo an toàn
* Kĩ thuật đảm bảo an toàn là kĩ thuật được thiết kế để phát hiện, ngăn ngừa hoặc loại bỏ tấn công
* Dịch vụ an toàn: bí mật; xác thực; toàn vẹn; sẵn sàng; chống chối bỏ
* Kĩ thuật an toàn: ngăn cản vật lí; định danh; cấp quyền; xác thực; mã hóa; ký số
  1. Giao thức an toàn mạng
* Giao thức mạng là tập hợp các quy tắc và quy ước điều khiển việc trao đổi thông tin giữa các hệ thống máy tính
* Giao thức an toàn mạng là một giao thức mật mã (áp dụng các phương pháp mật mã) được sử dụng để bảo vệ dữ liệu trên máy tính và dữ liệu truyền thông
* Các giao thức an toàn phổ biến:
  + Giao thức xác thực: PAP/CHAP; Kerberos; EAP
  + Giao thức ở tầng Application va Transport: S/MIME; SSH; SSL/TLS
  + Giao thức ở tầng Network: IPsec

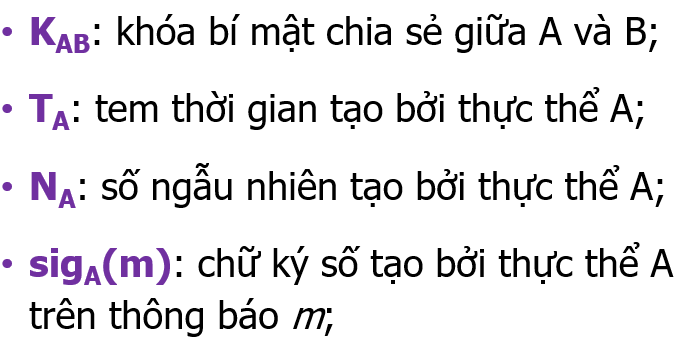
…..

Bài 2.1: Một số nguyên thủy xác thực

1. Khái niệm, phân loại xác thực và giao thức xác thực
   1. Xác thực

* Là hành vi xác nhận sự thật một thuộc tính của một đối tượng hoặc một chủ thể.
* Xác thực thông điệp là một cơ chế cho phép khẳng định rằng thông điệp không bị thay đổi trong quá trình truyền và bên nhận có thể kiểm tra được nguồn gốc của thông điệp
* Xác thực thực thể là một thủ tục mà qua đó, một thực thể thiết lập 1 tính chất được yêu cầu cho 1 thực thể khác
  + Thực thể: người dùng; tiến trình; client; server
  + Tính chất được yêu cầu: có mật khẩu đúng; có nắm giữ khóa bí mật;…
  1. Giao thức xác thực
* Là một giao thức mật mã với mục đích thực hiện xác thực
  1. Ký hiệu quy ước



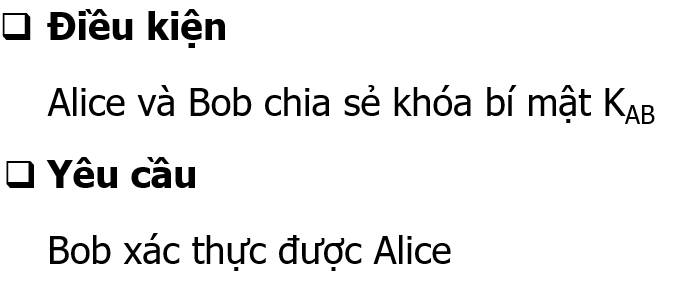


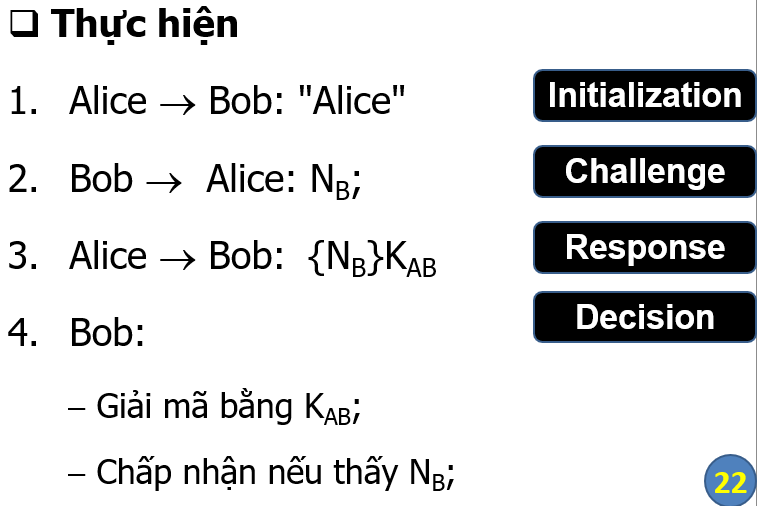
1. Truyền trực tiếp bí mật
   1. Điều kiện: Alice và Bob chia sẻ mật khẩu PA
   2. Yêu cầu: Bob xác thực được Alice
   3. Thực hiện:

* Alice -> Bob: “Alice”, PA
* Bob kiểm tra tính hợp lệ và chấp nhận hoặc từ chối truy cập
* Chú ý:
  + Mật khẩu có thể bị chặn thu
  + Nhưng thực tế vẫn được sử dụng rộng rãi (phần lớn các website)
  + Để đảm bảo an toàn, quá trình xác thực được thực hiện qua một kênh mã hóa
  1. Phương án lưu trữ mật khẩu của Alice ở phía Bob:
* Lưu trữ dạng rõ
* Lưu trữ dạng băm
* Lưu trữ dạng băm có phụ gia

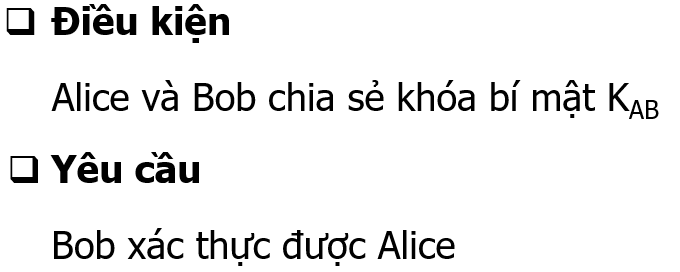
1. Thách đố bằng nonce
   1. Khái niệm

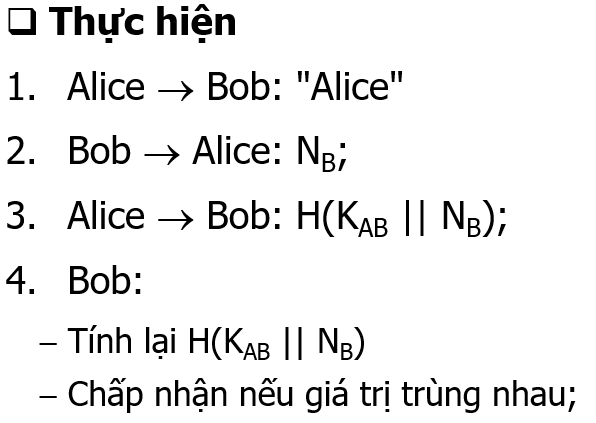
* Nonce là một đại lượng được sinh ngẫu nhiên, sao cho không bao giờ gặp lại
* Chú ý:
  + Để không lặp lại, nonce thường có kích thước lớn (128bit)
  + Việc nonce không lặp lại giúp đảm bảo tính tươi của thông điệp (thông điệp chỉ được sử dụng 1 lần) trong xác thực
  + Nonce luôn được sinh bởi bên xác thực (authentication server); xác thực 2 chiều thì cả 2 bên cùng sinh (bởi lúc đó cả 2 cùng là authentication server)
* Mã đối xứng + nonce



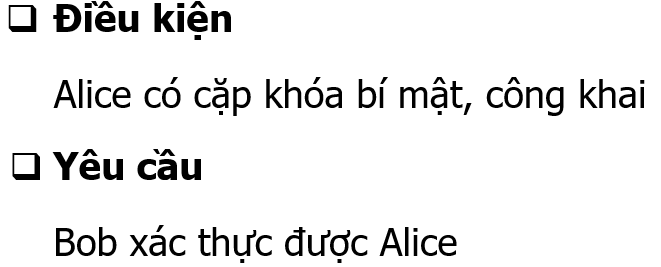


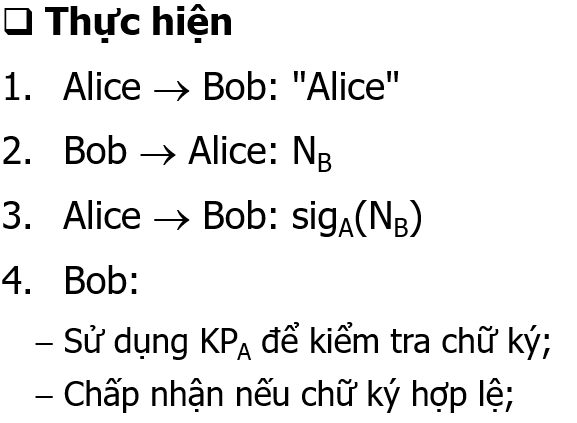
* Hàm băm + nonce



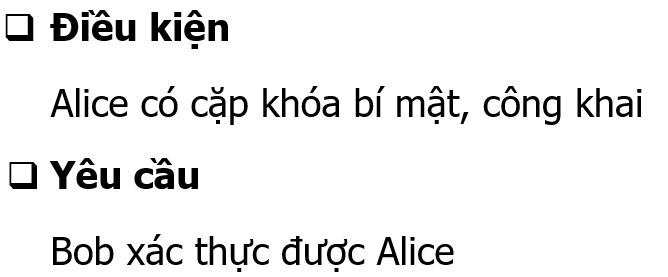


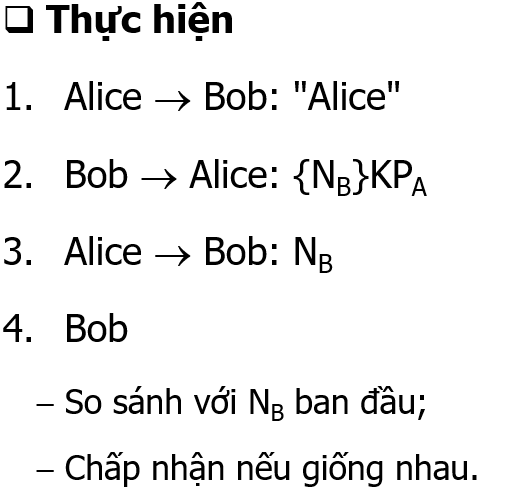
* Chữ ký số + nonce





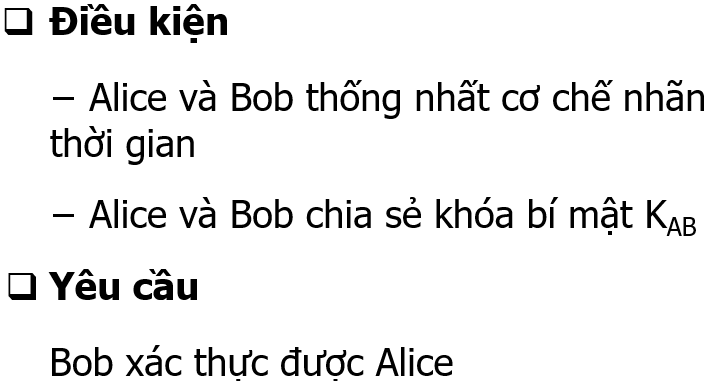
* Mã hóa khóa công khai + nonce

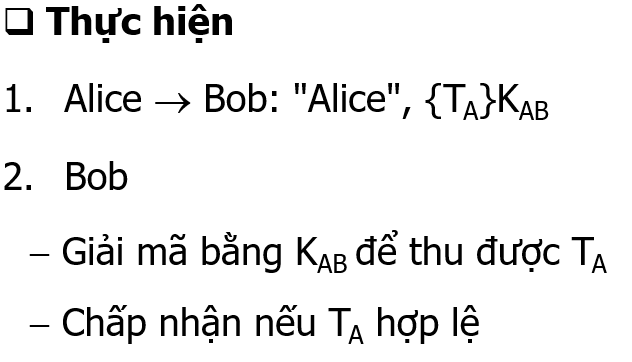




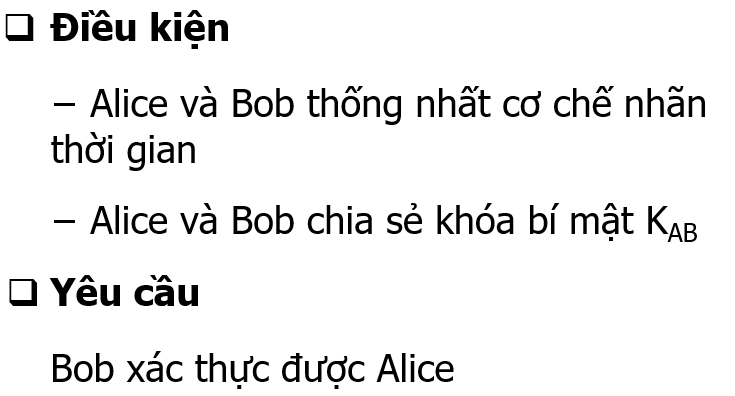
1. Thách đố bằng timestamp
   1. Khái niệm

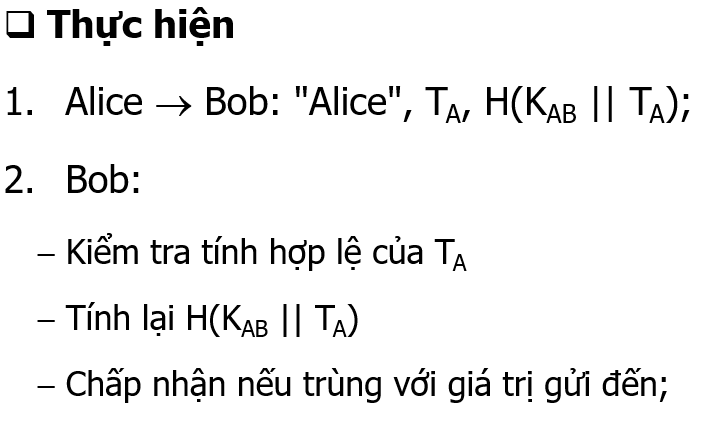
* Là đại lượng chỉ thời gian trên đồng hồ của mỗi bên tham gia xác thực
* Luôn có sự sai lệch giữa đồng hồ của các bên nên phải xác định một sai số chất nhận được
* Mỗi timestamp sẽ hợp lệ trong một khoảng thời gian nhất định để chống tấn công phát lại (replay attack) bên xác thực cần có cơ chế đảm bảo mỗi timestamp chỉ được sử dụng một lần (ví dụ: ghi nhớ timestamp gần nhất được sử dụng bởi mỗi thực thể)
* Mã đối xứng + timestamp



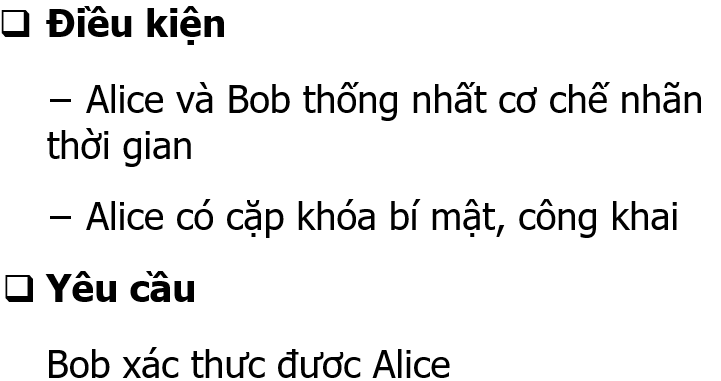


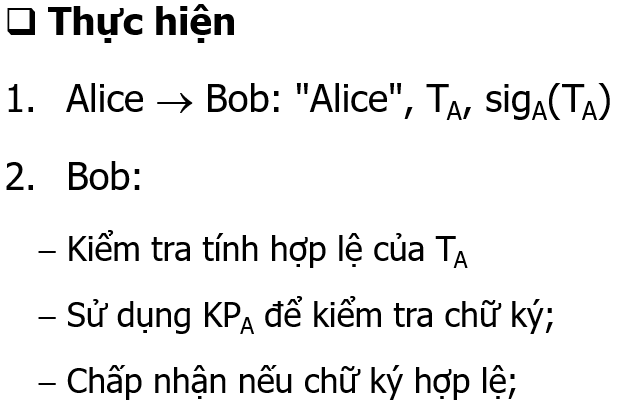
* Hàm băm + timestamp





* Chữ kí số +timestamp





Bài 2.2: Giao thức xác thực

* Link tham khảo: [tại đây](http://etutorials.org/Networking/Wireless+lan+security/Chapter+2.+Basic+Security+Mechanics+and+Mechanisms/Authentication+and+Identity+Protocols/)

1. Một số thuật ngữ

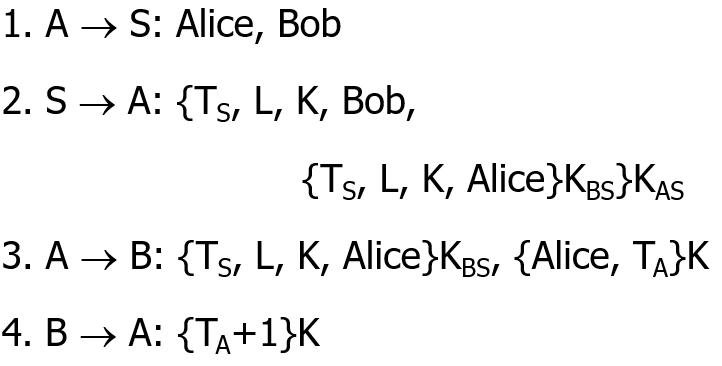
* Supplicant (hoặc Peer): bên được xác thực
* Authenticator: bên xác thực
* Authentication Server (AS): máy chủ xác thực
  + Cung cấp dịch vụ xác thực cho Authenticator
  + Giúp Authenticator xác thực Supplicant
* Network Access Server (NAS): máy chủ truy cập
  + Có thể là một authenticator
  + Cung cấp dịch vụ cho supplicant

1. Giao thức PAP và CHAP
   1. Tổng quan

* PAP và CHAP là 2 giao thức xác thực được sử dụng trong giao thức PPP
  + PAP (Password Authentication Protocol) truyền mật khẩu dạng rõ
  + CHAP (Challenge Handsharke Authentication Protocol) sử dụng cơ chế thách đố, giải đố
  1. Giao thức PAP
* Mã Protocol là 0xC023
* Là giao thức bắt tay 2 bước.
* Quá trình bắt tay:
  + Được thực hiện sau thiết lập liên kết (trong PPP) hoàn tất
  + Authenticate request (type code: 01) được gửi đi gồm thông tin Peer-ID và password.
  + Nếu kiểm tra trên cơ sở dữ liệu và đúng sẽ trả lại gói tin Authenticate Ack (type code: 02)
  + Nếu kiểm tra trên cơ sở dữ liệu và sai sẽ trả lại gói tin Authenticate Nak (type code: 03)
* Vấn đề an toàn
  + Không phải là phương thức xác thực mạnh
  + Mật khẩu được truyền ở dạng rõ
  + Có thể bị chặn và thu được thông tin trên đường truyền
  1. Giao thức CHAP
* Mã Protocol là 0xC223
* Là giao thức bắt tay 3 bước
* Quá trình bắt tay:
  + Authenticator gửi 1 challenge (code: 01) bao gồm: mã định danh (ID); số ngẫu nhiên; hostname của Authenticator
  + Tính toán giá trị băm của số ngẫu nhiên bằng hàm băm 1 chiều
  + Sau tính toán sẽ gửi lại 1 Response (code: 02) bao gồm: ID được mã hóa (encrypted version of the ID); mật khẩu sau khi băm với số ngẫu nhiên; số ngẫu nhiên; tên của thiết bị gửi response
  + Sau khi nhận được response, nó sẽ dựa trên tên từ Response để tìm kiếm mật khẩu tương ứng trên cơ sở dữ liệu. sau đó tính toán giá trị băm từ mật khẩu lấy được và số ngẫu nhiên từ challenge
  + Nếu các giá trị khớp nhau sẽ nhận được gói tin Success (code: 03) hoặc nếu giá trị không khớp sẽ nhận gói tin Failure (code: 04)
* Vấn đề an toàn: mật khẩu đã được băm trên đường truyền nên sẽ gây khó khăn hơn trong quá trình chặn và thu thông tin trên đường truyền

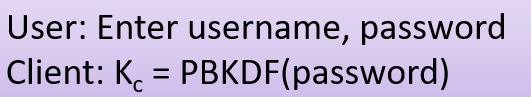
1. Giao thức Kerberos
   1. Thông tin chung về Kerberos

* Là giao thức mật mã dùng để xác thực trong mạng máy tính hoạt động trên đường truyền không an toàn
* Mục tiệu: xác thực 2 chiều trong mô hình client-server
* Dựa trên giao thức Needham-Shroeder
* Sử dụng mật mã đói xứng; có bên thứ 3 tin cậy là Key Distribution Center (trung tâm phân phối khóa)
* Là giao thức Single Sign-On (SSO): là 1 phương pháp xác thực mà người dùng chỉ cần đăng nhập 1 lần vào hệ thống và sau đó có thể truy cập nhiều ứng dụng và dịch vụ trọng hệ thống đó mà không cần đăng nhập lại
* Có nhiều phiên bản 1;2;3 và 4;5
  1. Giao thức Needham-Shroeder
* Điều kiện ban đầu
  + Alice và Bob cùng tin tưởng Sandy
  + Alice và Sandy chia sẻ khóa KAS
  + Bob và Sandy chia sẻ KBS
* Yêu cầu
  + Alice và Bob xác thực lẫn nhau, đồng thời thiết lập khóa chia sẻ K
* Thực hiện

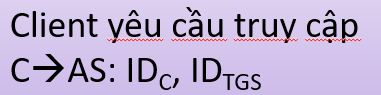


**Lưu ý:** L là thời gian tồn tại của thông điệp

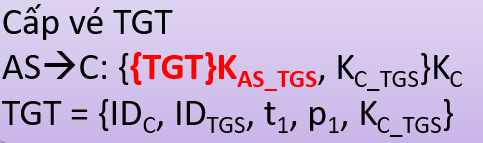
* 1. Giao thức Kerberos
* Từ viết tắt
  + AS: Authentication Server
  + TGS: Ticket Granting Server
  + KDC (=AS+TGS): Key Distribution Center
  + SS: Service Server
  + TGT: Ticket Granting Ticket
  + ST: Service Ticket
* Nguyên lý chung:
  + Luôn có 1 KDC
  + Khi client cần truy cập SS thì sẽ gửi yêu cầu đến KDC. AS sẽ là nơi nhận
  + AS sẽ cấp cho client 1 TGT
  + TGT giúp client truy cập đến TGS để yêu cầu truy cập 1 dịch vụ cụ thể
  + Kiểm tra quyền và cấp cho ST
  + Sử dụng ST để yêu cầu SS phục vụ
  + SS sẽ đáp ứng yêu cầu
* Chi tiết các thông điệp
  + khi người dùng nhập tên người dùng (username) và mật khẩu (password), client sẽ sử dụng mật khẩu đó để tạo ra một khóa bằng cách sử dụng hàm PBKDF (Password-Based Key Derivation Function). PBKDF là một loại hàm băm được thiết kế để tạo ra một khóa (key) từ một mật khẩu (password).



* + client yêu cầu truy cập: IDC là định danh client. IDTGS là định danh của TGS nơi client muốn nhận TGT

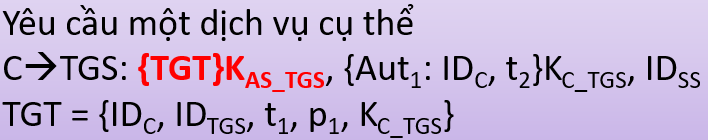


* + AS sẽ cấp vé TGT:

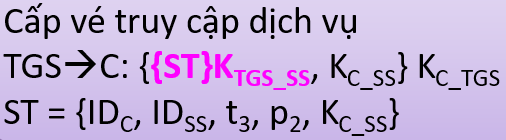


**Lưu ý:** p là thời gian tồn tại của thông điệp

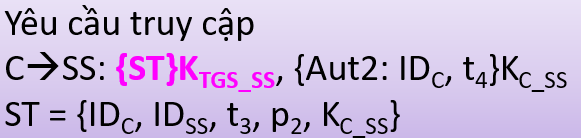
* + Từ client gửi TGT đến TGS:



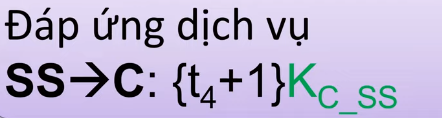
* + TGS phản hồi lại client:



* + Client gửi đến SS:



* + Đáp ứng dịch vụ:



* SSO trong kerberos: chỉ cần qua AS lần đầu và các lần sau đó có thể gửi đến trực tiếp TGS
  1. Độ an toàn
* Theo lý thuyết là khá an toàn, tuy nhiên kerberos chỉ cung cấp dịch vụ chứng thực, do đó không ngăn được quản trị viên cấp giấy phép cho user sử dụng trái phép hoặc lựa chọn mật khẩu yếu hoặc máy chủ trung tâm ngừng hoạt động

1. Giao thức EAP; 802.1X và RADIUS
   1. EAP (Extensible Authentiaction Protocol)

* Giao thức xác thực (khả) mở rộng
* Được mô tả trong RFC 3748
* Thường được sử dụng trong kết nối mạng không dây và trong kết nối điểm-điểm.
* Hoạt động ở tầng DataLink nên EAP truyền tin mà không cần IP
* Không cố định phương thức xác thực
* Phương thức xác thực được xác lập trong quá trình xác thực
* Cho phép tùy chọn phương thức xác thực phù hợp với yêu cầu về an toàn
* Cho phép thay đổi phương thức xác thực với sự thay đổi tối thiểu trong phần cứng, phần mềm
  1. Định dạng gói tin (code)
* 1 Request
* 2 Response
* 3 Success
* 4 Failure
  1. Các phương thức xác thực
* Có khoảng hơn 40 phương thức
* Nổi bật: MD5 (type=4); TLS (type=13); TTLS (type=21); PEAP (type=21); FAST (type=43); IKEv2 (type=49); …
  1. Quá trình kết nối
* Bước 1: Client yêu cầu kết nối tới NAS (chưa phải là giao thức EAP)
* Bước 2: NAS gửi yêu cầu định danh client
* Bước 3: Client gửi phản hồi định danh
* Bước 4: NAS gửi một thách thức (tùy thuộc vào phương thức)
* Bước 5: Client gửi phản hồi gửi lại phiên bản đã băm của thách thức
* Bước 6: NAS kiểm tra đúng sai và cho phép hoặc từ chối truy cập
  1. RADIUS
* Giao thức RADIUS là một công cụ quan trọng trong việc xác thực người dùng khi họ cố gắng truy cập vào mạng hoặc các dịch vụ mạng từ các kết nối từ xa
* Giao thức hoạt động theo mô hình client-server. Trong đó, RADIUS client hay NAS tương tác với RADIUS server thông qua một hay nhiều RADIUS proxies
* Được thiết kế dựa trên nền tảng kiến trúc AAA (authentication-authorization-accounting)
* RADIUS hỗ trợ nhiều phương thức xác thực khác nhau
* Cơ chế hoạt động kết hợp với xác thực: NAS như nơi dung chuyển còn thực hiện xác thực sẽ trên RADIUS server
  1. 802.1X: là chuẩn quy định sử dụng EAP qua môi trường LAN (xác thực người dùng truy cập vào mạng LAN)

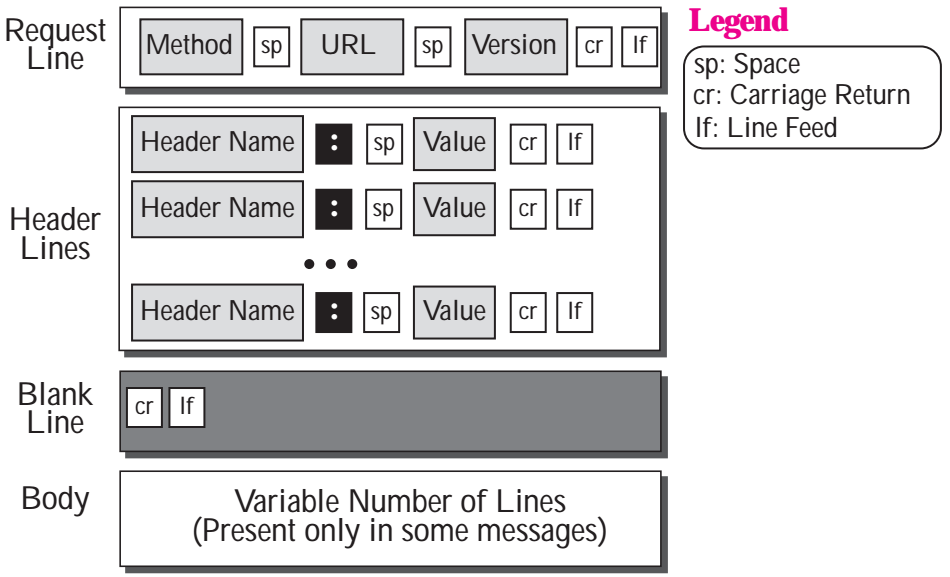
Bài 3: các vấn đề an toàn ở tầng ứng dụng

1. Giao thức tầng ứng dụng

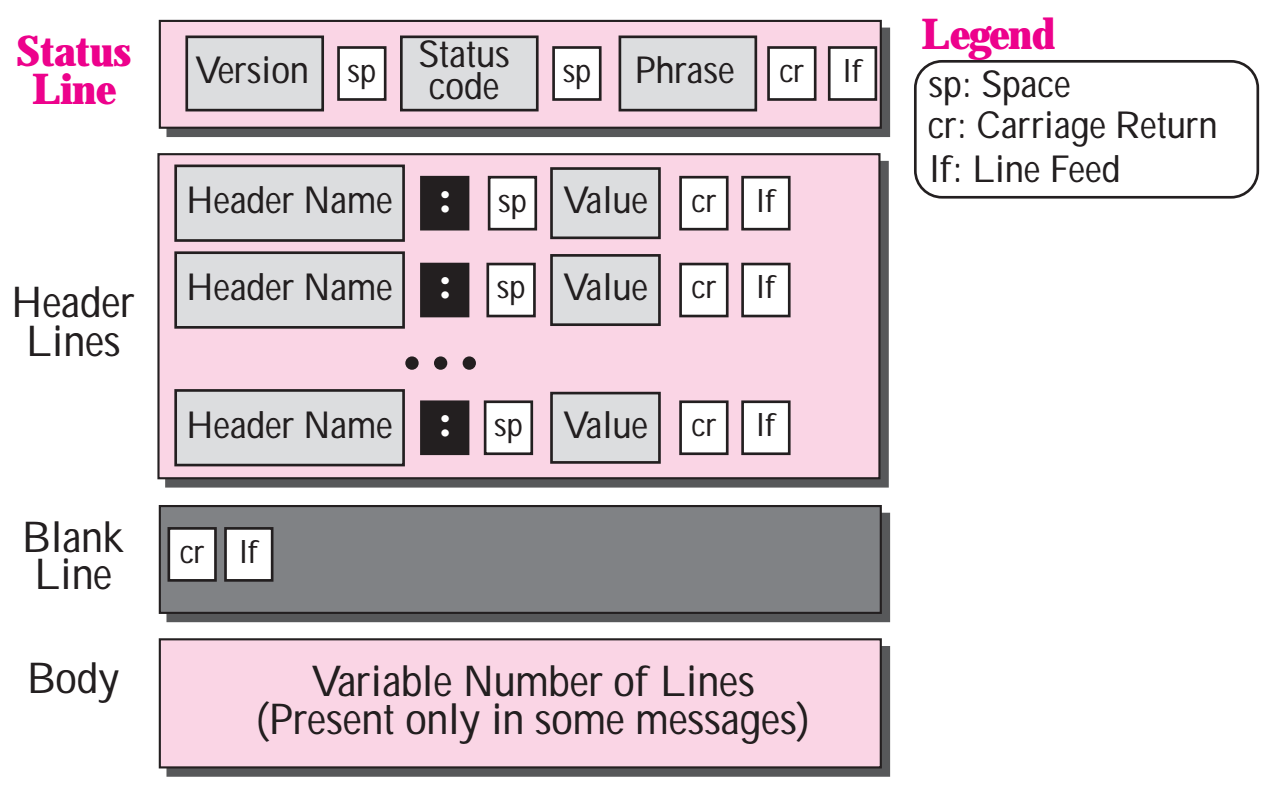
* DHCP
* DNS
* Remote login: TELNET; SSH
* File Transfer: FTP; TFTP.
* World Wide Web: HTTP
* Mail: SMTP; POP; IMAP; MIME
* Network Management: SNMP

1. World wide web
   1. Giao thức HTTP

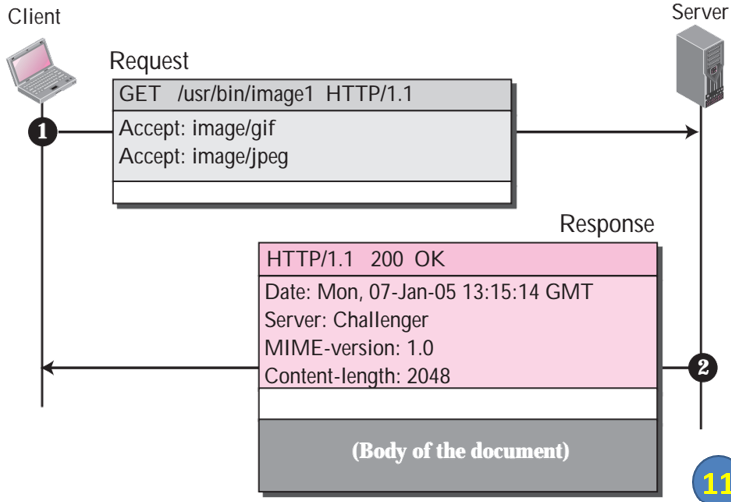
* HTTP = HyperText Transfer Protocol
* Port: 80
* Là giao thức chủ đạo của web
* Phiên bản:
  + Ver 0.9, 1.0: thuần túy văn bản
  + Ver 1.1: hỗ trợ “application/octet-stream”
  + Ver 2: chủ yếu là binary
* Hoạt động của HTTP dựa trên quá trình gửi yêu cầu từ client đến server và nhận lại phải hồi
* Không trạng thái: mỗi yêu cầu từ client đều độc lập và không phụ thuộc vào các yêu cầu trước đó
  1. HTTP Request



* 1. HTTP Response



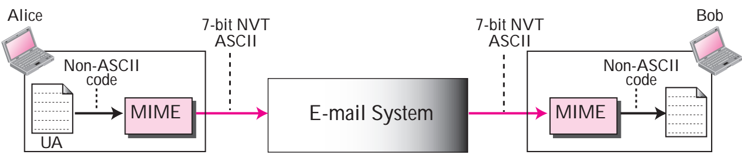
* 1. Ví dụ



* 1. Vấn đề an toàn HTTP
* Client không thể xác thực Server
* Client có thể được xác thực bởi:
  + Web server: basic, digest, … hiếm khi được sử dụng
  + Web application: usename/password (không thuộc phạm vi HTTP)
* Ngoài một vài cơ chế xác thực, HTTP không cung cấp dịch vụ an toàn nào khác (bí mật, toàn vẹn)
* Cần hỗ trợ của giao thức an toàn lớp dưới. HTTP + TLS/SSL = HTTPS (port 443)

1. Thư điện tử
   1. Tổng quan về thư điện tử

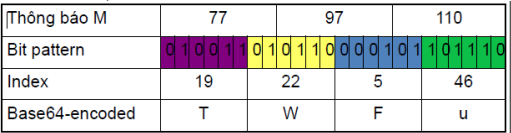
* User agent (UA): phần mềm hoặc ứng dụng dùng để tạo, gửi , đọc và quản lý email
* Mail transfer agent (MTA): là thành phần trong máy chủ thư điện tử, nhằm cung ứng các dịch vụ thư điện tử
* Mail delivery agent (MDA): là thành phần trong mail server nhận email từ MTA được sử dụng để nhận và lưu trữ gửi đến mailbox tương ứng của người dùng
* Soạn thảo:
  + UA soạn thảo email cung cấp các thông tin cần thiết như chủ đề; nội dung; địa chỉ email của người nhận
* Chuyển đổi sang dạng chuẩn:
  + Sau khi soạn thảo, email sẽ được chuyển đổi sang định dạng chuẩn được xác định bởi RFC 822
* Phân chia thành phần:
  + Email sau khi chuyển đổi được chia làm 2 phần: tiêu đề (header) và body
* Gửi thư:
  + UA kết nối đến MTA trên mail server. MTA nhận thông tin từ UA và chuyển giao đến email của người nhận
* Nhận thư:
  + MTA của người nhận nhận được MTA của người gửi và MDA tiếp nhận và lưu trữ
  + UA của người nhận kiểm tra email trên email server
  + Hiển thị email cho người nhận
  1. SMTP
* Là giao thức tầng ứng dụng được sử dụng để truyền và gửi email từ người gửi đến với mail server của người nhận
* Là giao thức truyền thông điệp có hiệu quả cao
* Port: 25
* Thủ tục làm việc mỗi phiên:
  + Khởi tạo phiên
  + Khởi tạo client
  + Truyền thư
  + Kết thúc phiên và đóng kết nối
* Các trạng thái của SMTP
  + Khi client gửi 1 lệnh SMTP tới server, client nhận về 1 mã trạng thái cho máy gửi biết điều gì đang xảy ra
  + Với thông điệp có gắn 3 con số ở đầu dòng thể hiện từng trạng thái riêng
* Hạn chế của SMTP
  + Chỉ sử dụng với dữ liệu dạng ASCII 7 bit
  + Không có cơ chế xác thực
  + Thông điệp gửi đi không được mã hóa
  + Dễ bị tổn thương (bởi spam, mất định danh người gửi)
* Giải pháp
  + SMTP mở rộng
  + MIME
  1. MIME
* Là một chuẩn internet về định dạng thư điện tử
* Hầu hết mọi thư điện tử Internet đều được truyền qua SMTP theo định dang MIME
* MIME cho phép gửi các thông điệp trên internet không chỉ thuần văn bản có thể gửi mail kèm ảnh, link, …
* Thêm trong header khai báo MIME: MIME version; method mã hóa dữ liệu; …
* Biến đổi dữ liệu non-ASCII sang ASCII



* 1. Phương pháp mã hóa base64
* Chia file nhị phân thành nhóm nhỏ dài 3 byte
* Mã hóa từng nhóm 3 byte thành 4 kí tự ASCII 7bit in ấn như sau:
  + Gộp 3 byte thành 24bit chia thành 4 nhóm 6 bit có giá trị từ 0-63
  + Mỗi nhóm 6 bit tương ứng với 1 kí tự in ấn như sau:



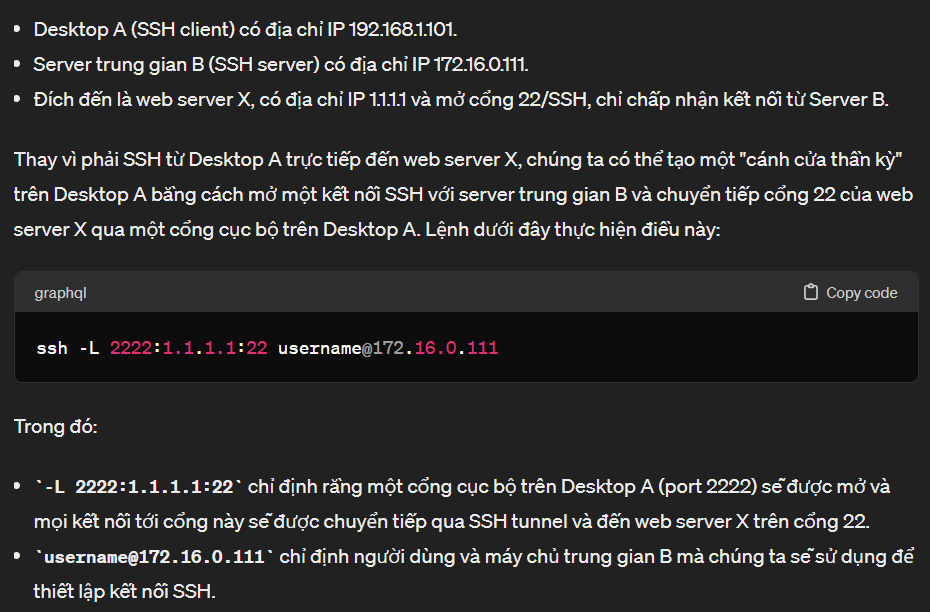
* + Ví dụ:



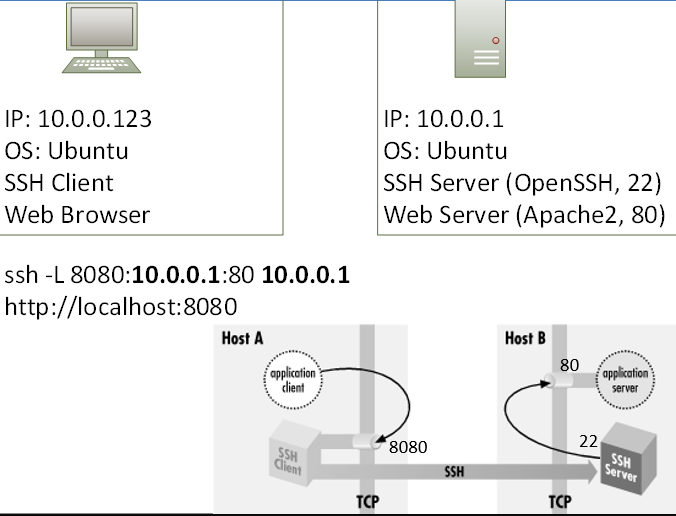
* 1. POP3
* Là giao thức ở tầng ứng dụng
* Dùng để truy cập và lấy thư điện tử từ mailbox trên máy chủ thư tín
* Port: 110
* Trước POP3 có 2 phiên bản là POP và POP2
* Hoạt động của POP3:
  + POP3 client xác thực thành công với server, server có khóa và mở được maildrop (sử dụng để xử lý email trên email server) thích hợp. client có thể truy cập đến mailbox trên server để kiểm tra và nhận thư
  + Nếu maildrop không mở, server đóng kết nối hoặc client gửi lệnh xác nhận và bắt đầu lại từ đầu
  + Thiết lập kết nối ở port 110 và server có thể truyền với giao thức POP3
  + Client gửi lệnh quit khi xong công việc để kết thúc phiên làm việc
  + Server từ trạng thái Transaction chuyển sang update. ở transaction vẫn có thể nhận email tuy nhiên quá trình xử lý mail được hoãn lại
  + Server gửi goodbye tới client để chính thức kết thúc phiên làm việc
  1. IMAP
* Là giao thức tầng ứng dụng
* Là giao thức cho phép client truy cập và quản lý mail trên mail server. Nó cho phép user trực tiếp thực hiện các thao tác trên mailbox (bằng cách đồng bộ).
* Port: 143
* Khác biệt với POP3: IMAP thích hợp cho người dùng muốn truy cập email từ nhiều thiết bị và muốn quản lý email trên máy chủ, trong khi POP3 thích hợp cho người dùng muốn lưu trữ email trên thiết bị cá nhân và không cần đồng bộ hóa giữa các thiết bị.
  1. Vấn đề an toàn
* Trong SMTP không có yêu cầu xác thực giải pháp là ESMTP (port: 587)
* nếu một người dùng hợp lệ trên domain.com có quyền gửi email từ bất kỳ định danh nào (ví dụ: [user1@domain.com](mailto:user1@domain.com), user2@domain.com, v.v.), thì họ có thể gửi email từ bất kỳ định danh nào đó điều này có thể gây ra rủi do giả mạo tấn công cách phòng chống áp dụng các biện pháp bảo mật như SPF, DKIM (DomainKeys Identified Mail), và DMARC (Domain-based Message Authentication, Reporting, and Conformance) để xác minh tính hợp lệ của các email gửi từ tên miền của họ.
* Với POP3 và IMAP nguyên bản thì khi truyền đi sẽ là dạng rõ
* Vì vậy update lên sẽ dùng SSL/TLS
  + SMTPS (port 465)
  + POP3S (port 995)
  + IMAPS (port 993)
  1. S/MINE
* Là một tiêu chuẩn bảo mật cho việc gửi và nhận email trên Internet
* Nó dử dụng MIME để mở rộng giao thức email và cung cấp tính năng bảo mật
* S/MIME sử dụng mã hóa bất đối xứng và PKI (lưu trữ khóa công khai của người gửi và người nhận) để bảo vệ email
* Mã hóa email: sử dụng public key của người nhận để mã hoad email. Người nhận sử dụng private key của mình để giải mã
* Xác thực email: người gửi sd khóa bí mật để ký email, người nhận sử dụng public key của người gửi để xác thực chữ kí

1. Đăng nhập từ xa

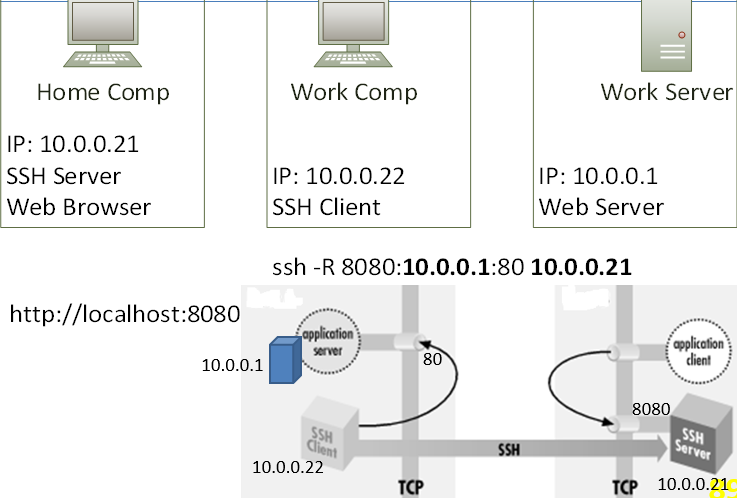
* Chú ý: ngoài đăng nhập từ xa, có thể truyền file và dữ liệu
  1. TELNET
* Có 2 thành phần:
  + Telnet client (máy user)
  + Telnet Server (máy ở xa)
* Hoạt động:
  + User khởi chạy telnet client trên terminal của mình
  + telnet client chuyển câu lệnh của user tới telnet server
  + telnet server thực thi cậu lệnh người dung trên máy ở xa và gửi kết quả tới client
  + telnet client hiển thị kết quả lên terminal
* telnet hoạt động trên TCP port 23; xác thực bằng tài khoản trên hệ điều hành
* telnet hoàn toàn không dùng mật mã
  1. SSH
* Secure Shell
* Ssh có vai trò như telnet nhưng sinh ra để thay thể telnet
* Hiện nay SSH-2 (RFC4251) là giao thức quản trị từ xa mặc định của mọi admin
* Chức năng an toàn của SSH:
  + Xác thực thực thể 2 chiều:
    - Xác thực SSH server: public key (RSA)
    - Xác thực SSH user: password hoặc public key
  + Xác thực thông điệp đảm báo tính toàn vẹn: HMAC (MD5 hoặc SHA1)
  + Mã hóa dữ liệu: mã hóa đối xứng
  + Nén dữ liệu (tùy chọn): GZIP
* Kiến trúc giao thức SSH:
  + SSH-TRANS:
    - Xác thực server, trao đổi khóa, đảm báo bí mật, toàn vẹn cho gói tin, nén dữ liệu, …
    - Client mở kết nối TCP tới server (không phải là 1 phần của SSH-TRANS
    - Một chuỗi định danh được client gửi đến server và server gửi đến client được sử dụng trong xác thực server
    - Thỏa thuận bộ thuật toán và tham số: client đề xuất nhiều, server lựa chọn 1 bộ. bộ thuật toán: trao đổi khóa, mã hóa, xác thực, nén
    - Thực hiện trao đổi khóa, server được xác thực trong quá trình trao đổi khóa
    - Báo hiệu việc kết thúc trao đổi khóa. Hai bên đã có khóa chung để mã hóa thông điệp
    - Client yêu cầu dịch vụ: SSH-AUTH; SSH-CONN
  + SSH-AUTH: xác thực client, thực thi trên SSH-TRANS
  + SSH-CONN: thiết lập và quản lý kết nối giũa ssh và server. Ghép nhiều kết nối logic vào 1 kết nối SSH. Thực hiện trên SSH-TRANS sau SSH-AUTH
  + SSS Application: triển khai các ứng dụng an toàn trên SSH. Sử dụng giao thức SSH-CONN
* SSH port forwarding
  + SSH tunnel hiểu như 1 kênh truyền
  + Có 3 loại SSH port forwarding:
    - Local port forwarding: là dạng kết nối từ ssh client được chuyển tiếp qua ssh server rồi đi đến host/server đích
    - Remote port forwarding: kết nối từ ssh server được chuyển tiếp qua ssh client, rồi đi đến host/server đích
    - Dynamic port forwarding: tương tự local port forwarding, kết nối từ SSH client được chuyển tiếp qua ssh server rồi đến địch tùy ý không định trước
  + Local port forwarding:
    - Mở kết nối ssh đến ssh server với tùy chọn “-L port:host:hostport”
    - Ví dụ:



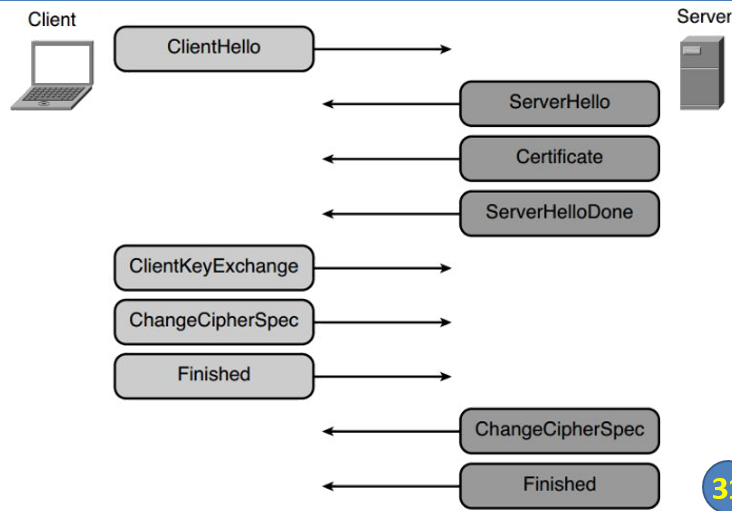
* + - SSH server nằm cùng máy với Application Server và Application client nằm cùng SSH client. Cho phép A client kết nối đến A server thông qua cổng trên SSH client
    - Ví dụ:



* + Remote port forwarding
    - Application Server nằm cùng máy với SSH Client, Application Client nằm cùng máy với SSH Server, Cho phép Application Client kết nối với Application Server thông qua cổng trên SSH Server
    - Ví dụ:



1. Bộ giao thức SSL/TLS
   1. Tổng quan về SSL/TLS

* Lịch sử phát triển
  + SSL 1.0 -> 2.0 -> 3.0 -> TLS 1.0 -> TLS 1.1 -> TLS 1.2 -> TLS 1.3
* Các dịch vụ của SSL/TLS
  + Xác thực thực thể (1 chiều hoặc 2 chiều)
  + Thỏa thuận bộ tham số mật mã
  + Trao đổi khóa phiên
  + Nén dữ liệu
  + Bí mật dữ liệu
  + Xác thực (và toàn vẹn) dữ liệu
* Giao thức SSL: gồm 4 giao thức con
  + SSL handshake protocol: thực hiện chức năng thỏa thuận các thuật toán, tham số, trảo đổi khóa, xác thực server và client (nếu có lựa chọn)
  + SSL record protocol: phân mảnh, nén, tính mac , mã hóa dữ liệu, chịu trách nhiệm đóng gói các message sẽ truyền đi
  + SSL alert protocol: thông báo lỗi
  + SSL change cipher spec protocol: thông báo xác nhận kết thúc giao đoạn Handshake protocol
  1. Giao thức Record (slide không rõ)
  2. Giao thức Handshake
* 
* Quá trình “Hello” giữa client và server thu được:
  + Danh sách thuật toán mật mã
  + Thuật toán băm
  + Thuật toán nén dữ liệu
* Server –> client certificate: server gửi các chứng thư số của nó cho client, client kiểm tra tính hợp lệ và chấp nhận
* Server ->client serverhellodone: server đã gửi hết tất cả các thông tin mà nó có cho client
* Client -> sever clientkeyexchange: client sinh là một pre\_master\_key (48 byte), mã hóa nó bằng public key của server về sau server gải mã được thì server coi như xác thực, thuật toán mã hóa đã thương lượng; server giải mã bây giờ cả client và server đều có PMK, RC, RS -> master key -> session keys
* Client -> server ChangeCipherSpec: thông báo từ lúc này mọi trao đổi đều được mã hóa bằng thuật toán và session key đã thương lượng
* Client -> server Finished để kết thúc quá trình thiết lập SSL tunnel
* Server gửi 2 gói tin tương tự đến client
* Trường hợp server xác thực thêm client:
  + Server: Trước khi gửi gói tin ServerHelloDone, Server yêu cầu Client gửi Certificate
  + Sau ServerHelloDone Client gửi chứng thư số của mình cho Server
  + Sau ClientKeyExchange Client CertificateVerify: H = Hash(tất cả các message đã trao đổi trước đó), Vc = Ký lên H; Server CertificateVerify: H = Hash(tất cả các message đã trao đổi trước đó), Vs = Kiểm tra chữ ký của Client lên H

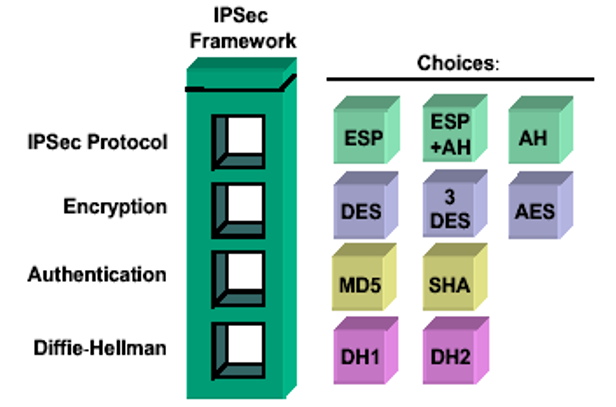
Bài 4.1: Giới thiệu về VPN

1. Tổng quan về VPN
   1. Mạng riêng (private network)

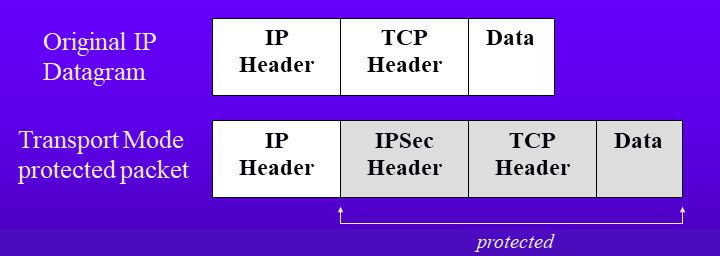
* Là mạng được xác định, vận hành bởi cá nhân tổ chức cho mục đích riêng
* Là một tổ hợp trang bị thiết bị mạng tạo thành 1 vùng mạng riêng biệt dưới sự kiểm soát của chủ sở hữu
* Sử dụng dải IP private: 10.0.0.0/8; 172.16.0.0/12; 192.168.0.0/16
  1. Mạng riêng ảo (VPN)
* Là sử dụng mạng công cộng làm cơ hạ tầng để truyền thông tin nhưng vẫn đảm bảo là một mạng riêng (toàn quyền quản trị, cô lập lưu lượng, sử dụng dải địa chỉ IP private)
* VPN không phải là giao thức, nó có thể hiện thực hóa bằng các giao thức khác nhau
* VPN có thể sử dụng mật mã hoặc không
* Phân loại
* Dịch vụ an toàn có thể cung cấp
  + Đảm bảo tính bí mật
  + Tính toàn vẹn
  + Tính xác thực
  + Chống tấn công phát lại
* Lợi ích:
  + An toàn
  + Chi phí thấp
  + Nâng cao khả năng kết nối
  + Nâng cao khả năng mở rộng
  + Sử dụng hiệu quả băng thông

1. Giới thiệu về IPsec
   1. Tổng quan về IPsec

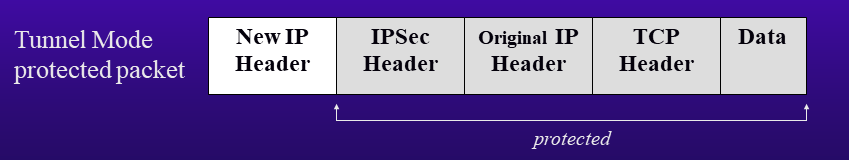
* Internet protocol security
* Sử dụng để triển khai VPN
* Được phát triển bởi IETF
* Thực hiện việc an toàn các gói IP
* Cung cấp khả năng:
  + Xác thực nguồn gốc thông tin
  + Kiểm tra tính toàn vẹn thông tin
  + Đảm bảo bí mật nội dung thông tin
  + Cung cấp khả năng tạo và tự động làm tươi khóa mật mã một cách an toàn
* IPsec cung cấp khung an toàn tại tầng 3 của mô hình OSI
* Khung giao thức:



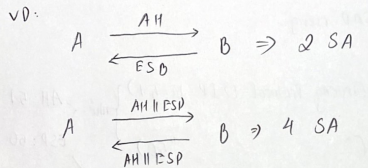
* IPsec cung cấp an toàn cho 3 tình huống:
  + Host to host
  + Host to gateway
  + Gateway to gateway
* IPsec hoạt động ở 2 chế độ:
  + Chế độ transport (end to end)
  + Chế độ tunnel (gateway to gateway) (cho VPN)
* Transport mode: bảo vệ dữ liệu các lớp trên



* Tunnel mode: bảo vệ các gói tin gốc (gồm dữ liệu các lớp trên và IP header)



* Quy trình hoạt động của IPsec
  + Ban đầu xác định luồng lưu lượng cần bảo vệ
  + Bước 1: hai bên thiết lập kết nối an toàn bằng cách thoả thuận một Security Association (SA1). SA1 thường bao gồm các thông tin về phương thức xác thực, thuật toán mã hoá và độ dài khóa.
  + Bước 2: Thiết lập kênh truyền thông an toàn và xác thực đối tác: Sau khi SA1 được thiết lập, một kênh truyền thông an toàn được tạo ra giữa hai bên và họ xác thực lẫn nhau sử dụng các thông tin từ SA1.
  + Bước 3: Pha IKE thứ 2: Trong bước này, hai bên tiếp tục qua trình giao thức IKE để thoả thuận một IPSec SA (SA2) trên kênh an toàn vừa được thiết lập. SA2 cung cấp các thông tin cần thiết để thực thi các giao thức bảo mật IPsec, chẳng hạn như ESP (Encapsulating Security Payload) hoặc AH (Authentication Header), cùng với các thuật toán mã hoá, xác thực và khóa.
  + Bước 4: Thực thi AH hoặc/và ESP: Với SA2 đã được thiết lập, các giao thức bảo mật IPsec như AH hoặc ESP được sử dụng để bảo vệ dữ liệu truyền qua kênh an toàn. Các thuật toán mã hoá, xác thực và khóa được chỉ định trong SA2 sẽ được sử dụng để thực hiện quá trình bảo vệ này.
  + Bước 5: Các khóa được sử dụng trong quá trình mã hoá và giải mã dữ liệu sẽ được lưu trữ trong SAD (Security Association Database) cho mỗi kết nối
  + Kết thúc: IPsec sẽ bị xóa
* Pre-Shared Key trong IPsec (PSK): là phương thức xác thực bổ biến trong IPsec, chỉ dùng cho xác thực không dùng cho mã hóa
  1. Tổ hợp an toàn SA
* Security associations
* Là một khái niệm cơ bản của bộ giao thức IPsec
* Là một kết nối logic theo 1 hướng duy nhất giữa 2 thực thể sử dụng IPsec
* Có 2 kiểu SA
  + IKE SA
  + IPsec SA
* Một SA gồm 3 phần:
  + SPI
    - Là 1 trường 32 bit dùng để xác định 1 SA để gắn với 1 gói dữ liệu
    - Là 1 chỉ số duy nhất gắn cho mỗi bản ghi của cơ sở dữ liệu SAD
    - Được định nghia bởi người tạo SA, được lựa chọn bởi hệ thống địch khi thương lượng SA
  + Destination IP Address: địa chỉ IP của node đích
  + Security protocol: mô tả giao thức an toàn IPsec được dùng an toàn có thể là AH hoặc ESP
* Nội dung của 1 SA:
  + Giao thức an toàn: AH, ESP
  + Thuật toán, khóa mật mã: DES, 3DES
  + Phương pháp, khóa xác thực cho AH | ESP
  + Thông tin liên quan đến khóa
  + Thông tin tin liên quan đến chính SA
* Tính đơn hướng của SA
  + Với 2 điểm liên lạc: cần một SA cho mỗi hướng
  + SA có thể cung cấp các dịch vụ an toàn cho một phiên VPN (được bảo vệ bởi AH hay ESP)



* Cơ sở dữ liệu cho SA
  + Cơ sở dữ liệu tổ hợp an toàn – SAD
    - Thời gian hiệu lực của SA
    - Thông tin về AH và ESP (các khóa, các thuật toán, …)
    - Chế độ transport hoặc tunnel
  + Cơ sở dữ liệu chính sách an toàn – SPD
    - Xác định lưu lượng cần bảo vệ
    - Các mục policy để thiết lập SA
* Xử lý outbound: IP packet -> qua SPD -> SAD -> xử lý IPsec -> SPI và IPsec packet -> gửi tới B
* Xử lý inbound: từ A đến B -> SPI và packet -> sử dụng SPI để tham chiếu đến SAD gỡ bỏ SAD và SPD thành IP packet
  1. Giao thức AH
* AH được thêm header vào gói tin IP
  + Xác thực người gửi: tiêu đề này dùng cho việc xác thực gói dữ liệu IP gốc tại người nhận là ai gửi
  + Toàn vẹn gói tin: tiêu đề này cũng giúp nhận biết về bất cứ sự thay đổi nào về nội dung của gói dữ liệu
  + AH không thực hiện mã hóa
* Transport mode
  + AH được chèn vào sau IP header trước TCP|UDP header
  + Không tạo header mới
* Tunnel mode
  + Một gói tin IP khác được thiết lập dựa trên gói tin IP cũ
  + Tạo một tạo 1 IP header mới: tiêu đề này chứa địa chỉ nguồn và đích được sử dụng để định tuyến trên Internet. Địa chỉ IP nguồn và đích trong tiêu đề này thường là địa chỉ của các thiết bị đầu cuối của VPN (như các IPsec gateway).
  + Tiêu đề IP cũ (bên trong) chứa địa chỉ nguồn và đích
  + AH nằm giữa
* …
  1. Giao thức ESP
* Là giao thức đóng gói tải an toàn của IPsec
* Đảm bảo tính:
  + Toàn vẹn
  + Xác thực
  + Bí mật (mã hóa)
* …