# Chương VII ĐIỀU KHIỂN TRUY NHẬP

## 7.1 Khái niệm cơ bản

*Xác thực* (authentication) là pha đảm bảo an toàn đầu tiên mà hệ thống cần kiểm soát khi người sử dụng mới đăng nhập, nhằm đảm sự chính danh,

*Điểu khiển truy nhập* (AC: access control) là pha thứ hai quyết định xem người dùng có thể làm gì và như thế nào trong ngôi nhà hệ thống này.

## 7.2 Ma trận điều khiển truy nhập là gì?

Ma trận điều khiển truy nhập (Access Control Matrix – ACM) là một công cụ hình thức cơ bản để thể hiện trạng thái bảo vệ hệ thống một cách chi tiết và chính xác. Nó sẽ cung cấp thông tin chi tiết và chính xác rằng, tại thời điểm đang xét, một tài nguyên nào đó có thể được truy nhập bởi một NSD nào đó với những quyền cho phép cụ thể xác định nào đó. Cụ thể là, mô hình được đặc trưng bởi bộ ba (S,O,R) trong đó:

* S={s1,s2, …, sn}: tập hợp các chủ thể (subjects) có thể yêu cầu truy nhập đến tài nguyên, ví dụ như NSD (users) hay các tiến trình kích hoạt bởi NSD
* O={o1,o2, …, om}: tập hợp các đối tượng truy nhập (objects) tức là các tài nguyên, phổ biến là các tệp dữ liệu lưu trữ.
* R={r1,r2, …, rk}: tập các quyền cụ thể xác định sẵn mà mỗi phần tử của S có thể có đối với mỗi phần tử của O

## 7.3 Các giải pháp để cài đặt ACM

Các giải pháp để cài đặt ACM một cách khả thi đều dựa trên nguyên tắc chung là phân rã ma trận để tiện lưu trữ và truy xuất đồng thời biểu diễn các thành phần này bằng các khái niệm biểu hiện (đối tượng quản lý của hệ điều hành) thích ứng với phạm vi mới. Cụ thể có các giải pháp phổ biến sau:

* Phân rã theo cột: tạo nên đối tượng quản lý là các danh sách điều khiển truy nhập (access control list: ACL). Các ACL sẽ được gắn vào các đối tượng tài nguyên (object), cung cấp danh sách các NSD và quyền có thể truy nhập đến đối tượng
* Phân rã theo dòng: tạo nên các danh sách khả năng (capability list), được gắn với các chủ thể (NSD), cung cấp danh sách các tài nguyên mà chủ thể có thể sử dụng với quyền truy nhập cụ thể tươn ứng.
* Thông qua các biểu diễn gián tiếp khác, ví dụ như khóa, nhóm, vai trò, …

Tất cả các giải pháp này đều cố gắng tạo ra một môi trường hoạt động có ngữ nghĩa sử dụng thuận tiện nhất.

## 7.4 Danh sách quyền truy nhập (Access Control List: ACL)

Danh sách quyền truy nhập (ACL) sẽ được gắn vào dữ liệu điều khiển của mỗi tài nguyền (v/d tệp dữ liệu). Chẳng hạn, dữ liệu điều khiển của một file F sẽ được gắn một danh sách các truy nhập có thể như (U:r,w,o; V:w; S:r); qua đó, hệ thống cho phép một chủ thể U có quyền làm chủ, được đọc và sửa lên F, và chủ thể V được sửa, chủ thể S được đọc đối với tệp F.

## 7.5 Danh sách năng lực (capabilility list)

Đây là cách tiếp cận của việc phân hoạch ma trận theo dòng, từng là theo chủ thể (subject). Tài khoản của mỗi chủ thể sẽ chứa một cấu trúc dữ liệu để lưu tất cả các quyền truy nhập tài nguyên mà chủ thể này có, tức là một danh sách năng lực truy nhập (capability list). Danh sách truy nhập này cần phải được tạo ra nhỏ nhất có thể, vừa đủ có thể làm việc theo đúng chức năng của chủ thể -- đây chính là nguyên lý khá phổ biến trong CNTT, với tên gọi nguyên lý tối thiểu đặc quyền (principle of least priviledge)

Bên cạnh hai tiếp cận phổ biến hơn nói trên, người ta cũng đề xuất các phương án khác. Chẳng hạn như ACT (Access Control Triples), tức là danh sách các bộ ba (chủ thể, đối tượng, quyền truy nhập) được lưu trong một cấu trúc bảng; nó chính là biểu diễn rút gọn của ma trận toàn thể bằng cách triệt tiêu toàn bộ các ô dữ liệu trống. Cách tiếp cận khác sử dụng các khái niệm riêng như Lock và key: các tài nguyên có cấu trúc điều khiển gọi là lock mà chủ thể nào muốn sử dụng thì phải có key tương ứng (cũng là một thông tin điều khiển). Cách tiếp cận này phối hợp cả hai kiểu sử dụng ACL (danh sách truy nhập) và CL (danh sách năng lực).

## 7.6 Mô hình harrison-ruzzo-ullman và điều khiển truy nhập tùy nghi

Điều khiển truy nhập tùy nghi (Discretionary Acess Control - DAC) là sự thể hiện của nguyên lý: quyền truy nhập (right) cho từng cặp (chủ thể, đối tượng) có thể được xác định riêng rẽ và có thể quyết định bởi chủ thể chủ nhân của đối tượng (owner). Nguyên lý này là đối lập với nguyên lý điều khiển dựa trên chính sách chung của hệ thống mà ta sẽ nói tới khi bàn về mô hình điều khiển cưỡng chế (Mandatory Access Control – MAC) ở mục tiếp theo. Trước hết ta hãy làm quên với một mô hình mang tính hình thức cao mang tên 3 tác giả đã đề xuất nó, tức là mô hình HarrisonRuzzo-Ullman (viết tắt là HRU). Mô hình này có thể coi là một phiên bản mang tính hình thức (formal model) của DAC.

## 7.7 Mô hình Harrison-Ruzzo-Ullman (HRU)

Mô hình HRU là một cố gắng khái quát hóa (hình thức hóa) các khái niệm về trạng thái bảo vệ và ma trận truy nhập, hướng tới mô tả các hoạt động của hệ điều hành và tính an toàn của nó. Nhờ có mô hình này, các tiếp cận cụ thể về điều khiển truy nhập có thể được đặc tả dễ dàng hơn, các chính sách và thuộc tính về ATBM có thể được đặc tả chính xác hơn. Đặc biệt bài toán An toàn (Safety problem) đã được hình thành trên cơ sở mô hình và việc đánh giá một hệ thống có đang ở trạng thái an toàn hay không là có thể biết (quyết định) được.

Như đã nói ACM xác định trạng thái bảo vệ hiện thời của hệ thống (protection state). Một hệ thống được quan niệm là an toàn nếu như trạng thái hiện thời của nó nằm trong khu vực an toàn, xác đinh bởi một tập các trạng thái an toàn Q mà người thiết kế chính sách mong muốn. Nếu ta gọi P là tập tất cả các trạng thái mà hệ thống có thể đạt đến thì P\Q chính là tập các trạng thái không an toàn. Tuy nhiên rõ ràng người ta không thể liệt kê hết các trạng thải của Q (vì quá lớn) nên chỉ có thể mô tả nó bằng các đặc tính cần thiết. Mỗi một hành động của hệ điều hành (do sự vận động của hệ thống) như sự thực hiện của 1 lệnh sẽ tạo nên một chuỗi các chuyển dịch trạng thái. Bài toán Safety problem được đặt ra như là vấn đề làm sao để đánh giá một chuỗi dịch chuyển trạng thái có an toàn hay không, tức là có đưa trạng thái hệ thống thay đổi chỉ trong Q hay chạy ra ngoài P\Q.

Hiện nay người ta chưa giải quyết triệt để bài toán nói trên mà mới chỉ có những kết quả cục bộ, có thể đánh giá được tính an toàn trong một số điều kiện nào đó. Tức là bài toán thuộc loại ra quyết định này (an toàn hay không) chỉ mới giải được khi đưa về những trường hợp đặc biệt với một số điều kiện đủ tốt. Cách tiếp cận chính ở đây là qui các biến đổi trạng thái về một dạng chuẩn nào đó, tiện lợi để đánh giá chúng. Người ta đã chứng minh được rằng mỗi chuyển dịch (phát sinh từ một lệnh hệ điều hành) đều có thể được chuyển về một chuỗi gồm các thao tác nguyên tố cơ bản, tác động lên ACM. Nhờ đó việc đặc tả chuỗi chuyển dịch và đánh giá tính an toàn của chúng có thể thực hiện được. Các thao tác cơ bản này là như sau:

* Tạo mới: **create subject** *s*; **create object** *o*
  + Việc tạo mới mỗi chủ thể hay đối tượng nay sẽ tương ứng tạo mới một dòng hay cột của ACM  Xóa bỏ: **destroy subject** *s*; **destroy object** *o*
  + Sẽ xóa bỏ dòng/cột tương ứng của ACM
* Cấp quyền: **enter** *r* **into** *A***[***s***,** *o***]**
  + Thêm vào quyền *r* cho chủ thể *s* đối với đối tượng *o*.
* Thu quyền: **delete** *r* **from** *A***[***s***,** *o***]**
  + Thu hồi quyền *r* khỏi chủ thể *s* đối với đối tượng *o*

Như vậy một tiến trình p khởi tạo một tệp dữ liệu mới f với các quyền read, write cho nó, sẽ có thể viết dưới dạng một lệnh gồm một chuỗi các thao tác nguyên tố như sau:

**command** *create•file*(*p*, *f*)

**create object** *f***;**

**enter** *own* **into** *A***[***p***,** *f***];**

**enter** *r* **into** *A***[***p***,** *f***];**

**enter** *w* **into** *A***[***p***,** *f***];**

**end**

Một ví dụ khác, việc cấp phát quyền làm chủ cho một tiến trình *p* đối với tệp *g* được biểu hiện **command** *make•owner*(*p*, *g*)

**enter** *own* **into** *A***[***p***,** *g***];**

**end**

## 7.8 Điều khiển truy nhập tùy nghi (Discretionary Access Control – DAC)

Điều khiển truy nhập DAC là một thể loại điều khiển truy nhập được sử dụng sớm và phổ biến nhất trong các hệ điều hành từ thời buổi sơ khai. Nó không có một định nghĩa chặt chẽ, chính xác vì không phải do một tác giả đưa ra mà hình thành một cách tự nhiên trong thực tế. Cho đến nay DAC vẫn là mô hình được ưa dùng phổ biến trong các hệ điều hành hiện đại. Đặc trưng gắn liền với nó là sự sử dụng khái niệm chủ nhân của mỗi đối tượng, tức là chủ thể có quyền cấp và kiểm soát khả năng truy nhập của các chủ thể khác đối với đối tượng này. Có thể thấy, mô hình này khá gắn bó với tiếp cận cài đặt ACL đối với ACM (sử dụng danh sách quyền truy nhập ACL). Bản thân quyền làm chủ cũng là một thứ quyền có thể cấp phát được. Do đó các quyền truy nhập có thể lan truyền trên các chủ thể.

Điều khiển truy nhập tùy nghi tạo nên sự linh hoạt mềm dẻo tối đa cho việc quản lý quyền truy nhập. Tuy nhiên sự phân tán cao độ của việc quản lý, cũng như sự cho phép dễ dãi trong việc cấp phát quyền, có thể tạo ra sự lan truyền quyền một cách không mong muốn, tức ra tạo ra những vấn đề an toàn bảo mật. Sự mềm dẻo dễ dãi này rất dễ bị khai thác, và hệ thống dễ bị tổn thương và không thể chống lại những nguồn và hình thức tấn công như: Trojan horse (con ngựa thành Troy), mã độc, lỗi phần mềm, NSD nội bộ có ý đồ xấu. Nguyên nhân chủ yếu như đã nói, hệ thống không thể kiểm soát được luồng thông tin (information flow) về điểu khiển truy nhập, do đó những kẻ chỉ là khách vãng lai hoặc vai trò thứ yếu trong hệ thống cũng có thể dần dần thu hoạch được những quyền truy nhập đối với những đối tượng quan trọng của hệ thống.

## 7.9 Điều khiển truy nhập cưỡng chế (mandatory access control – mac)

Ngược với DAC, Điều khiển Truy nhập Cưỡng chế (Mandatory Access Control – MAC), không cho phép các cá nhân chủ thể toàn quyền quyết định sự truy cập cho mỗi đối tượng mà cưỡng chế sự truy nhập tất cả các đối tượng theo một chính sách chung, được qui định bởi một cơ chế phân loại cấp bậc. Theo sự phân loại này các chủ thể được phân loại và được gán nhãn cấp bậc, thể hiện tầm quan trọng (đặc quyền) cao hay thấp trong hệ thống (xét trên phương diện an toàn bảo mật), và các đối tượng cũng được phân loại và gán nhãn thể hiện tính mật, tức là cần bảo vệ, cao hay thấp. *Cấp bậc* của chủ thể (security class) phải đủ cao thì mới có thể truy nhập được vào một đối tượng có một *nhãn bảo mật* mức nào đó (security clearance). Thông thường, *Cấp* của chủ thể cần phải không thấp hơn *Mức* bảo mật của đối tượng. Tóm lại, một luật truy nhập chung sẽ áp dụng để ra quyết định cho tất cả các yêu cầu truy nhập thay vì sự quản lý phân tán của các chủ nhân đối tượng như ở trong mô hình MAC.

Bên cạnh việc khống chế truy nhập thông qua cấp bậc của chủ thể và mức an toàn của đối tượng, một khái niệm cũng thường được sử dụng là sự phân nhóm theo thể loại thông tin. Thông tin trong hệ thống được phân loại theo các nhóm thể loại (cathegories), mà cũng được áp dụng cho cả chủ thể và đối tượng. Mỗi nhãn của mỗi chủ thể hay đối tượng sẽ có hai thành phần (cấp/mức, nhóm thể loại) trong đó nhóm thể loại được hiểu như một tập con của tập vũ trụ tất cả dạng các thông tin có thể có. Một cách khái quát, mỗi nhãn sẽ là một phần tử trong không gian tích đề-các (A,C) trong đó không gian của cấp/mức A có một quan hệ thứ tự đầy đủ trên đó còn không gian thể loại C là không gian các tập con có một dạng quan hệ thứ tự bán phần (tức là quan hệ tập con).

Có thể thấy luật truy nhập được xây dựng trên một quan hệ so sánh nhãn, mà hay được gọi là dominate tức là “chiếm ưu thế hơn” hay “cao hơn”. Một nhãn (*A*,*C*) là ưu thế hơn (dominate) nhãn (*A’*,*C’*) nếu và chỉ nếu *A*≥*A’* và *C**C’*. (Lưu ý rằng nếu dấu bằng xảy ra ở cả 2 chỗ thì cũng vẫn được chấp nhận.) Chú ý rằng, đã có một sự khái quát gộp chung lại của khái niệm cấp bậc của chủ thể và mức bảo mật của đối tượng thông tin. Chính vì vậy nhãn của chủ thể và nhãn của đối tượng thông tin có thể cùng đưa vào một không gian chung để so sánh và tạo nên tính đơn giản của qui luật truy nhập.

Để biểu diễn quan hệ “ưu hơn” trong một hệ thống thực tế, người ta có thể vẽ một đồ thị có hướng của các nhãn (như là nút đồ thị) mà các cạnh giữa chúng thể hiện quan hệ “ưu hơn” nếu có. Tuy nhiên để tránh phức tạp người ta có thể dấu không vẽ những cạnh thể hiện tính bắc cầu mặc dù hiển nhiên quan hệ “ưu hơn” là một quan hệ bắc cầu. Biểu diễn dạng đồ thị như vậy còn gọi là lưới. Mô hình thực tế BLT mà ta sẽ trình bày sau đây cũng dựa trên cơ sở lưới như vậy.

## 7.10 Mô hình Bell- LaPadula (BLP)

Đây là một mô hình phổ biến trong các lĩnh vực liên quan đến an ninh quốc phòng, theo tiếp cận chung MAC. Mô hình BLP chú trọng vào bảo vệ tính mật cao độ, truy nhiên vẫn hỗ trợ khả năng phi tập trung hóa, tức là không dồn toàn bộ kiểm soát và quản trị truy nhập về một nơi duy nhất. Một mặt, để đảm bảo tính cưỡng chế cao, toàn bộ các yêu cầu truy nhập phải đi qua một bộ phận kiểm soát gọi là BLP reference monitor. Bộ phận monitor này sẽ kiểm tra xem yêu cầu truy nhập này có thỏa mãn các luật bảo mật chung, nếu đáp ứng mới thông qua. Tuy nhiên cũng có những chủ thể đặc biệt được coi là đáng tin cậy, luôn được thông qua. Các cấp bậc/thang mức được sử dụng là tối mật (Top Secret – TS), mật (Secret – S), nội bộ (Confidential – C) và Còn lại (Unclassified – UC) . BLP cũng cho phép phối hợp cả hai dạng cơ chế cưỡng chế và tùy nghi, trong đó cơ chế sử dụng bộ kiểm soát (BLP monitor) sẽ đảm bảo cưỡng chế áp dụng bộ luật chung, còn cơ chế tùy nghi có thể được thêm vào sau khi một yêu cầu truy nhập đã đáp ứng bộ luật.

Bộ luật của BLP chỉ có 2 luật cơ bản, được phát biểu hết sức đơn giản. Luật thứ nhất được gọi là Bảo mật đơn giản (Simple Security Property – SSP), trong đó một chủ thể s sẽ chỉ được phép thực hiện thao tác *đọc* (*read*) đối với một đối tượng o nếu nhãn của *s* là ưu thế hơn nhãn của *o*. Luật này áp dụng cho tất cả các chủ thể (kể cả đáng tin cậy, trusted subjects). Luật này đơn giản là không cho phép chủ thể cấp dưới được đọc biết thông tin ở cấp cao hơn, nó có thể được tóm tắt bởi 3 từ đơn giản trong tiếng Anh: *No Read Up*. Khi được phối hợp với một cơ chế tùy nghi, nó sẽ được phát biểu như sau: chủ thể *s* được đọc đối tượng *o* khi và chi khi nhãn của s ưu thế hơn nhãn của o đồng thời *s* có được cấp phép *đọc* đối với *o*.

Luật thứ hai có cái tên đơn giản hơn nữa, \*-luật (\*-property), và một phát biểu dường như khá ngược đời: một chủ thể *s* chỉ được thực hiện thao tác *viết* (*write*) lên đối tượng *o* khi nhãn của *o* là ưu thế hơn nhãn của *s*. Tượng tự như với luật SSP, \*-luật này cũng có một cách nói đơn giản: No Write Down. Tại sao vậy? Có thể hiểu là luật này được đưa ra để nhằm tránh việc những chủ thể ở cấp cao hơn có thể tình cờ tiết lộ thông tin cùng cấp xuống chủ thể cấp dưới. Tuy nhiên luật này chỉ áp dụng với các chủ thể không được coi là tin cậy (untrusted subjects). Tương tự như với SSP, luật này cũng có thể mở rộng để phối hợp với cơ ché tùy nghi.

## 7.11 Điều khiển truy nhập dựa vai trò (role-based access control – rbac)

Thực tế ứng dụng của điều khiển truy nhập đã làm nảy sinh một tiếp cận thiết kế điều khiển truy nhập kiểu mới, có khả năng bám sát và phản ánh tốt hơn những đặc trưng khái quát của các hệ thống thông tin doanh nghiệp, đặc biệt là các hệ thống có nghiệp vụ riêng (ví dụ như doanh nghiệp ngân hàng tài chính). Theo tiếp cận này, việc cấp các quyền truy nhập, khai thác tài nguyên (permission) không trực tiếp hướng tới người sử dụng cuối mà hướng tới, lớp hay cụm những người sử dụng giống nhau trên phương diện nhiệm vụ, vai trò xử lý thông tin. Khái niệm mới *vai trò* (role) được đưa ra để khái quát tượng trưng cho một dạng, một lớp các nhiệm vụ xử lý tin. Dễ thấy trong một hệ thống doanh nghiệp đặc thù, người ta có thể đưa ra định nghĩa của một tập các vai trò cơ bản, bao phủ hết các dạng nghiệp vụ đặc thù mà mỗi người sử dụng có thể phải thực hiện. Tập các vài trò thường có kích thước nhỏ hơn tập người dùng cuối rất nhiều vì thường mỗi vai trò sẽ có một nhóm người dùng cuối được gán thuộc cho nó.

Như vậy ý tưởng cơ bản của tiếp cận mới này là sự định nghĩa của tập hợp các vai trò công việc cơ bản (thường mang nặng tính nghiệp vụ), tương ứng với mỗi vai trò là một dạng nhiệm vụ xử lý thông tin cơ bản, và việc ban phát quyền sử dụng, truy nhập tài nguyên đến các vai trò. Một người dùng cuối tuy không được ban phát quyền truy nhập một cách trực tiếp, nhưng vẫn được hưởng các quyền thích hợp do “ăn theo” những vai trò mà người dùng này được gán cho. Chú ý rằng một người dùng có thể có một hoặc nhiều vai trò khác nhau.

Mô hình mới này được goi là mô hình điều khiển truy nhập hướng vai trò (RoleBased Access Control – RBAC). Cách tiếp cận của nó rất phù hợp với mô hình doanh nghiệp có nghiệp vụ đặc trưng, vì vậy các khái niệm của nó rất gần với trực giác, bám sát được các yêu cầu về quản lý sử dụng tài nguyên phản ánh đúng trách nhiệm, quyền hạn và năng lực của các dạng vị trí (vai trò) công việc trong doanh nghiệp.

Chú ý rằng trong RBAC, sự gắn quyền vào vai trò (role-permission assignment) thường là lâu dài, trong khi đó ở DAC sự gắn quyền trực tiếp đến người dùng cuối (user-permission assignment) có thể mang tính ngắn hạn và thay đổi thường xuyên (không bám sát đặc thù công việc, mà bám vào nhu cầu cụ thể có thể thay đổi hàng ngày). Vì vậy RBAC thể hiện hàng loạt các ưu điểm vì phù hợp hơn với quản lý trong hệ thống thông tin doanh nghiệp. Hiển nhiên, nó có khả năng diễn tả cao các chính sách tổ chức của doanh nghiệp: phân công theo vai trò là cơ sở cho sự sự tách biệt các nhiệm vụ cũng như tạo ra cơ chế đại diện ủy nhiệm. RBAC cũng hỗ trợ khả năng đảm bảo đặc quyền tối thiểu hợp lý (Least previledge) và khai quá hóa thông tin dữ liệu (data abstraction). Đồng thời RBAC rất mềm dẻo và tiện lợi kinh tế cho việc đáp ứng nhanh các thay đổi về chính sách bảo mật. Một yêu cầu bảo mật mới sẽ chỉ dẫn đến thay đổi cách thức gán quyền truy nhập vào các vai trò, chứ không dẫn đến sự thay đổi cụ thể trực tiếp vào dữ liệu điều khiển người sử dụng.

Mô hình RBAC là độc lập với các mô hình DAC và MAC. Mô hình này là trung tính với chính sách (policy neutral): chính cách cấu hình các vai trò trong hệ thống sẽ xác định, thể hiện chính sách muốn áp đặt vào hệ thống. Không nên hiểu khái niệm *vai trò* (RBAC) là tương tự với *nhóm người dùng* (user group, RBAC). Nhóm người dùng đơn giản là một tập thể người dùng (cùng làm việc, hay cùng chia sẻ điều gì đó) nhưng mỗi người dùng vẫn có thể có các quyền khai thác khác nhau. Vai trò có thể coi là khái niệm trung gian giữa một tập các người dùng và một tập các quyền khai thác.

RBAC không phải là một mô hình đơn nhất mà được thực ra một tập hợp các mô hình phát triển ở mức độ khác nhau và có tính thừa kế. Đơn giản nhất là mô hình cơ bản RBAC0 (base model) mà mọi mô hình khác đều thừa kế các tính chất của nó. Mô hình RBAC1, mô hình có sự phân cấp vai trò (role hierarchy), và mô hình RBAC2, mô hình có ràng buộc (constrainsts) đều thừa kế trực tiếp trên cơ sở RBAC0. Ngoài ra mô hình RBAC3 (consolidated model) là mô hình cao cấp hơn cả, thừa kế cả RBAC1 và RBAC2.

## 7.12 Mô hình cơ sở RBAC0

Mô hình thể hiện sự tương tác vận động giữa tập U các người dùng (users), tập P các quyền truy nhập khai thác (permissions) và tập R các vai trò hay vị trí công việc. Tập S các phiên làm việc (sessions) thể hiện nhiều khả năng đăng nhập khác nhau có thể xảy ra của một người dùng mà có nhiều vai trò khác nhau. Các ánh xạ (assignments) thể hiện các tương tác và quan hệ giữa các tập này, qua đó cũng nói lên các chức năng của điều khiển truy nhập.

Ánh xạ UA  U x R (User Assignment) thể hiện sự gắn người dùng vào các vai trò. Một vai trò có thể được gắn cho nhiều người dùng và một người dùng cũng có thể có nhiều vai trò (loại mũi tên trong hình vẽ thể hiện mối nhiều - nhiều này). Tích Đềcác U x R thể hiện tập tất cả các cặp phép gắn giữa 1 NSD và 1 vai trò có thể có; vì vậy, quan hệ UA chính là một tập hợp con của tập tích Đề-các này. Khái niệm một người dùng có nhiều vai trò là phù hợp với thực tế khi có nhiều người có khả năng làm việc kiêm nhiệm, đặc biệt là trường hợp quản lý kiêm nhiệm và quản lý làm thay vai trò nhân viên (khi thiếu người).

Ánh xạ PA P x R (Permission Assignment) thể hiện sự gắn quyền truy nhập cho các vai trò. Đây cũng là quan hệ nhiều-nhiều: một vai trò thì có nhiều quyền truy nhập (nhiều tài nguyên khác nhau, hoặc cùng một tài nguyên với nhiều loại quyền khác nhau) và đương nhiên, cùng một quyền truy nhập có thể cấp cho nhiều vai trò khác nhau.

Tập S của các phiên (sessions) thể hiện quan hệ một-nhiều giữa các NSD (users) với các vai trò (roles). Tại một phiên làm việc, một người sử dụng có thể lựa chọn một hoặc nhiều hơn các vai trò, trong số các vai trò mà NSD đã được gán qua UA. Khi một phiên được tạo với nhiều hơn một vai trò (của cùng một người sử dụng), tập các quyền truy nhập có thể khai thác tại phiên này chính bằng hợp của các quyền truy nhập được phép đối với mỗi vai trò. Chú ý rằng mỗi NSD có thể cùng đồng thời mở ra nhiều phiên làm việc khác nhau (có thể ở trên nhiều máy). Trong mô hình hình thức, hàm Users: SU được mô tả như là hàm trên tập các phiên, cho đầu ra là 1 NSD là chủ của một phiên cho trước. Còn hàm Roles: S  2R được mô tả như hàm trên tập phiên mà cho đầu ra là tập các vai trò được gắn vào phiên (thông qua NSD chủ phiên). Thông qua các ánh xạ và hàm đã định nghĩa hình thức như trên, người ta có thể xác định được tập các quyền truy nhập ứng với mỗi phiên làm việc nào đó. Phiên nằm dưới điều khiển của NSD cho nên nó có thể được mở với bất kỳ tập vai trò là tập con của tập các vai trò đã gán cho NSD, và NSD cũng có thể thay đổi các vai trò này trong quá trình sử dụng phiên.

Chú ý rằng các quyền khai thác (permissions) chỉ được áp dụng cho các đối tượng dữ liệu tài nguyên chứ không áp dụng cho chính các đối tượng dữ liệu điều khiển truy nhập (theo mô hình RBAC). Chỉ có các đặc quyền của người quan trị mới có thể thực hiện sửa đổi cho các tập dữ liệu điều khiển U,R,S và P.

## 7.13 Mô hình cơ sở RBAC1

Mô hình này quan tâm đến sự tổ chức cấu trúc của các vai trò vốn dĩ không được xem xét trong RBAC0: mô hình này chỉ coi các vai trò như một tập độc lập, trong khi thực tế cho thấy điều ngược lại, đặc biệt ở các doanh nghiệp có hệ thống nghiệp vụ chuyên môn. Để phản ánh thực tế tốt hơn, mô hình RBAC1 đưa ra khái niệm tổ chức phân cấp các vai trò trong đó các vai trò ở cấp cao hơn có thể thừa kế sự sử dụng các quyền truy nhập của các vai trò cấp dưới. Sự phân cấp cao thấp này được đưa ra dựa vào một quan hệ bán thứ tự được định nghĩa bằng một cách nào đó. Quan hệ bán thứ tự là quan hệ thỏa mãn 3 tính chất: phản xạ, truyền ứng và bắc cầu. Sơ đồ mô hình có sự khác biệt duy nhất so với sơ đồ RBAC0 là ở chỗ đưa vào dấu hiệu của sự phân cấp các vai trò, bản chất cũng là một quan hệ nhiều - nhiều trên chính tập R. Quan hệ phân cấp có hình thức ký hiệu sau: RH  R x R, trong đó sử dụng ký hiệu ≥ cho quan hệ bán thứ tự nói trên.

## 7.14 Case study: điều khiển truy nhập trong hệ điều hành unix

Trong hệ điều hành Unix và các hệ điều hành phát triển thừa kế (như Linux), điều khiển truy nhập có một thiết kế đặc thù, có thể nói là tương thích với mô hình DAC và cài đặt ma trân truy nhập theo ACL (danh sách quyền truy nhập). Các khái niệm đối tượng quản lý của điều khiển truy nhập trong Unix là NSD (users), nhóm (user groups), tiến trình (processes) và tệp (files). Mỗi đối tượng chủ thể đều có định danh (identity – ID) duy nhất, tương ứng là UID, GID, PID (cho mỗi NSD, nhóm và tiến trình). Các đối tượng tài nguyên mà sự truy nhập được điều khiển là files và các thư mục.

## 7.15 Tổ chức của các file dữ liệu và dữ liệu điều khiển

Các file tổ chức theo một cấu trúc phân cấp của các thư mục. Bản thân các thư mục cũng được xem như các file đặc biệt. Mặc dù các thư mục được tổ chức phân cấp (dạng cây), quyền truy nhập các thư mục không có tính thừa kế. Chỉ có 3 quyền cơ bản để truy nhập dữ liệu file là đọc (Read), viết/sửa (Write) và thực hiên, chạy chương trình (Execute). Cũng 3 loại quyền truy nhập có thể áp dụng với thư mục (file đặc biệt) nhưng sẽ mang ý nghĩa thích hợp với thư mục:

* Đọc (Read): Xem danh sách các file trong thư mục
* Thực hiện (Execution): cho phép duyệt thư mục; chẳng hạn lệnh chuyển thư mục (chdir) sẽ yêu cầu quyền này mới có thể thực hiện được
* Kết hợp của Viết và Thực hiện sẽ cho phép tạo và xóa file trong thư mục đã cho
* Khi truy nhập một file theo đường dẫn đầy đủ: cần có quyền thực hiện trong tất cả chuỗi thư mục đi theo đường dẫn này.

Như đã nói, cài đặt của dữ liệu điều khiển truy nhập (ma trận truy nhập) là phỏng theo mô hình danh sách truy nhập ACL. Các quyền truy nhập của mỗi đối tượng tài nguyên (file) được cất vào một cấu trúc dữ liệu đi kèm với mỗi file, gọi là i-node (information node); cấu trúc này cũng còn lưu các thông tin thuộc tính khác của file. Tuy nhiên cấu trúc này là đơn giản và khái quát hơn nhiều nếu so với dự liệu cột của ma trận truy nhập. Nó không cho phép cấp phát quyền truy nhập đến từng NSD mà chỉ theo 3 lớp cơ bản: người chủ (NSD có quyền làm chủ - owner), nhóm chủ và tất cả các NSD khác.

Các quyền truy nhập một file (Đọc/Viết/Thực hiện) được lưu trữ bằng 3 bit (permission bits), trong khi hệ điều hành chỉ cấp phát quyền truy nhập theo 3 phạm vi khác nhau nói trên; vì vậy mỗi i-node của một file sẽ chứa 9 bit cho thông tin quyền truy nhập. Cùng với 1 bit lưu trữ xác định file này là file thường hay thư mục, chúng làm nên nhóm 10 bit thuộc tính cơ bản của file vẫn được thông báo theo mỗi dòng thông tin file khi gọi lệnh xem thư mục của hệ điều hành. Cụ thể là, khi được hiện thị, danh sách thuộc tính sẽ có dạng “drwxr-xr-x”, trong đó:

* Vị trí đầu tiên sẽ hiện thị “d”, nếu đối tượng này là thư mục, ngược lại là ký hiệu -
* Ba vị trí tiếp (nhóm bit đầu tiên) thể hiện các quyền mà người chủ có thể thực hiện với đối tượng; ba vị trí giữa thể hiện quyền của nhóm chủ; ba vị trí cuối thể hiện quyền của tất cả NSD còn lại (public)
* Trong mỗi chuỗi 3 vị trí, vị trí đầu cho biết về quyền đọc (r), quyền viết (w) hay quyền thực hiện (x); ở mỗi vị trí đó, nếu không có quyền thì sẽ hiển thị ký hiệu -

## 7.16 Chủ thể, sự đại diện và đặc quyền

Khái niệm các quyền truy nhập (permission bits) cho ta biết về khả năng có thể truy nhập vào một file của một NSD, tuy nhiên trên thực tế các chủ thể của hành động truy nhập lại là các tiến trình (chứ không phải NSD). Vì vậy sẽ có một cơ chế ngầm là tiến trình sẽ kiểm tra để biết NSD mà nó đại diện và lấy quyền truy nhập từ đó. Tuy vậy, có một vấn đề khó khăn đặc thù ở đây là: sẽ giải quyết ra sao nếu chương trình thực hiện (tiến trình) được xây dựng để thực hiện một *công việc nào đó mà đòi hỏi quyền truy nhập cao hơn quyền thực có của chủ thể* thực hiên nó?

Một ví dụ điển hình ở đây là chương trình ***passwd***, được xây dựng để giúp người sử dụng thay đổi mật khẩu của mình, và nó cần thực hiện cập nhật lên file lưu trữ các mật khẩu. Đây là dạng dữ liệu hệ thống, loại mà không thể được cho phép can thiệp bởi NSD thông thường, mà chỉ truy nhập được bởi chủ thể có đặc quyền quản trị hệ thống (root). Nhớ rằng dù sao thì *passwd* được tạo ra để cho người dùng thường sử dụng. Vì vậy Unix đã phải xây dựng thêm một cơ chế đặc biệt, nhằm giải quyết riêng khía cạnh đặc thù này của bài toán điều khiển truy nhập.

Cơ chế này cho phép mỗi tiến trình gắn với 3 định danh NSD (User ID) thay vì một duy nhất. Đó là: UID chủ (real UID, của người chủ thực hiện tiến trình), UID (effective UID) và UID lưu cất trạng thái trước (saved UID). Các quyết định về điều khiển truy nhập đều thông qua UID hiệu lực mà giá trị của nó thông thường đặt là UID chủ, tuy nhiên có thể thay đổi trong những trường hợp ngoại lệ, ví dụ như trường hợp sử dụng ***passwd*** nói tới ở trên. Khi có sự thay đổi như vây, UID lưu cất sẽ được sử dụng để lưu trữ UID cũ hơn vì nó sẽ được quay lại sử dụng sao này (giống như cơ chế ngăn xếp). Tương tự, mỗi tiến trình cũng sẽ được gán với 3 định danh nhóm tương ứng (GID chủ, hiệu lực và lưu cất).

Để giải quyết ngoại lệ nêu trên, Unix đưa ra một cờ (flag) gọi là setuid như một thuộc tính của tệp. Khi cờ này được đặt, tiến trình thực hiện sẽ có thể sử dụng đặc quyền cao hơn, mặc dù chỉ được gọi sử dụng bằng NSD mức thường. Cụ thể là khi được đặt, UID hiệu lực của tiến trình bị gọi sẽ được chuyển UID của người chủ của tệp (tạo ra chương trình) chứ không phải là người gọi thực hiện. Cụ thể quá trình một tiến trình sử dụng các UID của nó như sau. Khi một tiến trình tạo thông qua lệnh folk (tạo tiến trình con như một bản sao của tiến trình mẹ), tiến trình con này sẽ thừa kế cả 3 4 tiến trình mới bằng lệnh gọi thực hiện UID từ tiến trình mẹ. Khi một tiến trình gọi tạo một file (exec), nếu file này không đặt cờ setuid thì tiến trình tạo ra vẫn thừa kế 3 UID, nếu không (cờ setuid đặt) thì UID hiệu lực của tiến trình sẽ được đặt bằng UID chủ của file trong khi UID lưu cất sẽ giữ giá trị UID chủ (là UID hiệu lực trước đó).

Trường hợp ***passwd*** (và tương tự) sẽ được giải quyết cụ thể như sau. Tệp ***passwd*** là sở hữu của người quản trị hệ thống (đặc quyền cao nhất root) và được đặt cờ setuid (bit 1). Như vậy khi một tiến trình gọi thực hiện passwd, UID hiệu lực sẽ được đặt là root (chủ của passwd) trong thời gian tiến trình passwd hoạt động, và sẽ quay về UID trước khi passwd kết thúc. Nhờ đó mỗi NDS thường có thể đặt được mật khẩu của mình dù thao tác này liên quan đến việc cập nhật file hệ thống lưu trữ mật khẩu. Tuy nhiên cơ chế này cũng chính là một điểm yếu về an toàn cho hệ điều hành Unix nếu như cơ chế tạm mượn đặc quyền này bị lợi dụng.

4 Nhớ rằng việc một NSD gọi thực hiện một chương trình bản chất cũng là thông qua một tiến trình đang chạy (ví dụ: shell)

# Chương VIII AN TOÀN TRÊN INTERNET

## 8.1 Khái niệm an toàn với giao thức mạng

Thực tế của vấn đề an ninh mạng ngày nay đã cho thấy các vấn đề an ninh, các thể loại tấn công là rất đa dạng, liên quan đến hầu hết các giao thức khác nhau. Hầu hết các giao thức đều được thiết kế từ buổi sơ khai của Internet, tức là chỉ tập trung vào hoàn thành chức năng công việc, chưa chú ý đến an toàn bảo mật như ngày nay. Hơn nữa các giao thức mạng quan trọng đều đã được cài đặt rộng rãi phổ biến trên toàn thế giới, vì vậy sự thay đổi nâng cấp để giải quyết an toàn thông tin càng khó khăn. Một giao thức đã được phổ biến rộng khắp thì rất khó cho ai đó có thể can thiệp đến phần lõi, do sự cần thiết phải có tính tương thích với cộng đồng chung. Chính vì thế các giải pháp phải thiên về cải tạo và bổ sung thêm bên ngoài, càng làm hệ thống thêm phức tạp và dễ nảy sinh các vấn đề an ninh mới.

Như đã nói, chương này không có tham vọng khảo sát theo chiều rộng của lĩnh vực an toàn mạng, mà sẽ chủ yếu khảo sát trọng điểm mang tính điển hình. Trong mục này chúng ta sẽ khảo sát các dạng tấn công liên quan đến một giao thức mạng phổ biến là giao thức TCP. Sự khảo sát tấn công DOS đối với giao thức TCP là điển hình, không những do tính phổ biến (chiếm 90% trong các khảo sát thực tê), mà nó cũng thể hiện nguyên lý tấn công chung, cũng tương tự như các dạng tấn công DOS ít phổ biến với các giao thức như ICMP, UDP, DNS). Các mục tiếp theo (8.3-4) ta sẽ khảo sát một số công cụ bảo mật quan trọng đang được ứng dụng phổ biến.

## 8.2 Tầng giao vận và tấn công DOS bằng dòng thác SYN

Giao thức tầng giao vận TCP (Tranmission Control Protocol) cung cấp dịch vụ tạo ra truyền tin hướng kết nối điểm-điểm (point-to-point, connection oriented) giữa các tiến trình của người sử dụng. Nói một cách hình tượng, giao thức này cho phép 2 tiến trình ở hai máy khác nhau trên Internet có thể tạo ra một liên kết logic kiểu “đường ống” (pipeline), đảm bảo sự vận chuyển tin cây và đúng thứ tự của luồng các gói tin giữa 2 bên, giống như một luồng nước được bơm truyền qua đường ống. Mặc dù trên thực tế các gói tin được chuyển đi một cách độc lập rời rạc, bằng những đường liên kết vật lý (route) khác nhau, việc sử dụng các kỹ thuật phản hồi ghi nhận (acknowledgement), phát lại (retransmission), đồng hồ (timeout) … cho phép TCP tập hợp lại các gói tin theo đúng thứ tự, kiểm soát mất mát-phát lại, nhờ đó đảm bảo hình tượng “đường ống (một ví dụ điển hình về sự “nhân hóa”, dần hướng tới tầng ứng dụng).

Để khởi tạo liên kết đường ống này, một phiên làm viêc trên TCP sẽ được tạo ra bởi thủ tục bắt tay ba bước (3-way handsake), được minh họa qua hình vẽ 8.2. Thứ nhất, bên phát tin (Client) sẽ gửi đến bên nhận (Server) một gói tin có chức năng phát tín hiệu chào hỏi, gọi là gói tin SYN (bit cờ SYN của khối header được đặt). Gói tin này vừa báo danh và xin kết nối, vừa đề xuất một số thông số cho kết nối. Nếu chấp nhận kết nối, bên nhận (Server) sẽ gửi một phản hồi đồng ý, là gói tin loại SYN-ACK, và cũng đồng thời cho biết các thông số kết nối nó chấp nhận (thượng lượng giữa 2 bên). Nếu bên Client đồng ý với đề xuất thương lượng, thì phát gọi tin ACK, chính thức mở ra mối liên kết giữa 2 bên. Theo qui định chung của TCP, sự phản hồi ghi nhận cũng được thể hiện qua cung cách đánh số của các gói tin: một phản hồi ghi nhận sẽ kèm theo với số thứ tự của gói tin mà nó đang chờ đợi để nhận tiếp (tức là ghi nhận gói tin có số thứ tự ngay trước).

Trong hình vẽ 8.2 ở bên, có thể thấy: số thứ tự (Seq= sequence number) của gói tin SYN là x; khi nhận được nó, bên Server phản hồi bằng gói tin syn-ack, xác nhận đã nhận được gói tin số thứ tự x và đang chờ nhận gọi số x+1; gói tin phản hồi này có số thứ tự là y, do đó gói phản hồi ack của Client cũng xác nhân và báo đang chờ nhận gói y+1 kế tiếp.

Giao thức TCP được thiết kế từ thưở sơ khai của Internet, nên có thể nói theo tiêu chuẩn bây giờ là “ngây thơ” về an toàn thông tin. Bên server không hề kiểm tra xác thực bên client, tức là kiểm tra tính chân thật của sự tồn tại của Client cùng với danh tính của nó, mà máy móc đồng ý phục vụ, nếu tài nguyên còn đủ cho phép. Vì vậy đã tạo điều kiện cho kẻ địch có thể tấn công bằng cách “bắn phá” bên server bằng một dòng thác gói SYN. Các gói SYN gửi đến với số lượng cực lớn (có thể đạt đến hàng nghìn hay chục nghìn gói trong một giây) và đều có địa chỉ nguồn giả mạo. Vì vậy sự phản hồi của máy server sẽ chỉ đến những cái tai điếc (thực ra là đến các máy khác trên mạng không liên quan, và vì thế sẽ không hiểu và lọc bỏ các gói tin phản hồi syn ack này), còn bản thân địa chỉ nguồn tấn công thì đã được che giấu và rất khó phát hiện.

Cụ thể hơn, khi nhân một gói tin SYN, máy chủ Server sẽ máy móc kiểm tra tài nguyên và nếu còn đủ sẽ mở ngay một vùng bộ nhớ gọi là TCB (Tranmission Control Block), để đón chờ dữ liệu gửi tới, lưu tạm và xử lý. Trong thời gian chờ đợi thủ tục bắt tay ba bước chính thức kết thúc, vùng nhớ này vẫn luôn khóa lại để chờ, chỉ được giải phóng tới khi đạt đến thời gian chờ tới hạn (timeout) của một thủ tục bắt tay không thành công; trong khoảng thời gian chờ đợi gói ACK, máy server vẫn kiên nhẫn cách quãng phát lại một số gói SYN-ACK theo qui định của giao thức. Thời gian chờ này là khá lâu, tới 511 giây (đề phòng đáp ứng các tình huống xấu khi giao thông mật độ cao, có khả năng nhiều gói tin không tới đích kịp thời do tắc ngẽn), vì vậy TCB đã bị khóa lại không sử dụng trong một thời gian dài, gây tốn phí tài nguyên đáng kể.

Để đảm bảo máy chủ không bị cạn kiệt tài nguyên bộ nhớ do tình trạng mở TCB nhiều quá mức, các hệ thống máy chủ thường đặt ngưỡng số lượng TCB tối đa được mở và chờ đợi kết thúc bắt tay, chẳng hạn MAX-TCB-NO = 1024. Nếu kẻ địch tạo được một dòng thác SYN (SYN flood) với một tốc độ đủ cao đến mức, trong khoảng thời gian timeout qui định nói trên, số gói SYN đến Server là nhiều hơn cả ngưỡng MAX-TCB-NO, thì máy chủ có thể tự bảo vệ bằng cách từ chối không phục vụ các lời mời kết nối liên lạc tiếp theo (SYN), tức là bắt đầu từ chối dịch vụ đối với mọi liên kết tiếp theo, kể cả từ phía kẻ tấn công cũng như các khách hàng chân chính bình thường. Như vậy máy chủ đã bắt đầu đi vào trạng thái DOS (từ chối dịch vụ), tức là kẻ tấn công đã bắt đầu thành công.

Ở một góc độ khác, có những hệ thống máy chủ vận dụng một chiến thuật linh hoạt hơn, tức là khi đã đạt ngưỡng mở TCB thì không từ chối SYN mới đến mà chỉ đóng và giải phóng TCB đã chờ lâu nhất, để nhường chỗ phục vụ gói SYN mới. Tuy nhiên nếu dòng thác SYN đến quá lớn, việc giải phóng không chờ đời SYN cũ để phục vụ SYN mới sẽ diễn ra càng nhiều, càng liên tục, thời gian chờ đợi thực tế ngày càng thu hẹp lại, đến mức thậm chí không đủ để phục vụ một liên kết bình thường với một máy khách hàng chân chính. Tình trạng bão hòa này mới thực sự là phong tỏa thành công, máy chủ không hề còn khả năng xử lý các dịch vụ thông thường. Tuy nhiên để đảm bảo thành công mức độ này, kẻ tấn công sẽ phải tạo một dòng thác lớn hơn rất rất nhiều so với tình huống nói trước đó (khi máy chủ chỉ sử dụng chiến thuật từ chối SYN mới chứ không không hủy bỏ TCB cũ chưa tới hạn).

Để đảm bảo có một dòng thác lớn, lại tạo cơ chế che giấu cho mình, kẻ tấn công nham hiểm thường tổ chức tấn công theo kiểu phân tán (DDoS: Distributed DOS), bằng cách huấn luyện một lượng khá lớn các máy tính tay sai, bị điều khiển một cách vô thức, đồng loạt bắn phá máy chủ. Cơ chế huấn luyện tay sai thường được tổ chức thông qua việc tạo sâu độc, phát tán dần trên mạng, từ đó lén lút chiếm điều khiển bộ phận ở nhiều máy chủ. Sâu độc khi lan đến mỗi máy chủ, thì chủ động chạy lén lút, chiếm thời gian sử dụng CPU rất ít, nên khó bị phát hiện. Tận dùng thời gian thực hiện, sâu độc tìm cách mở rộng lan truyền sang các máy chủ lân cận (có kết nối trực tiếp với máy chủ nó đang cư trú lén), tiếp tục mở rộng tập máy chủ bị huấn luyến thành tay sai vô thức (vì lén lút khéo léo, nên quản lý hệ thống khó phát hiện được). Khi nhận tín hiệu phát động tấn công từ kẻ chủ mưu tấn công, toàn bộ các may tay sai sẽ đồng loạt cùng phát các luồng gói SYN đến máy chủ nạn nhân, tạo nên một dòng thác SYN cực lớn.

## 8.3 Một số giải pháp cho tấn công DOS trên TCP

Tấn công DOS là một dạng tấn công hết sức nguy hiểm, đến này vẫn chưa có biện pháp phòng vệ và trùy tìm kẻ tấn công thực sự chắc chắn thành công. Tấn công bằng dòng thác SYN đến giao thức TCP chính là dạng điển hình và phổ biến nhất của tấn công DOS. Ở đây chúng ta sẽ chỉ làm quen với một số giải pháp đơn giản, cơ bản.

* **Tối ưu hóa cấu hình máy chủ.** Đây là một biện pháp đương nhiên phải làm để có thể tăng sức đề kháng của máy chủ, tận dụng tối đa khả năng của nó. Cụ thể, ta có thể giảm thời gian tới hạn (timeout) để chờ phản hồi ACK từ Client xuống còn 10 giây (trước kia 511); tăng kích thước dòng đợi, tức số lượng TCB cùng mở cùng lúc; tháo bỏ các dịch vụ không thực sự hoạt động để huy động tập trung tài nguyên và làm giảm bề mặt tấn công.
* **Hợp tác đồng bộ các router trên toàn mạng Internet.** Các mạng con thành phần của Internet có thể liên hiệp và giúp đỡ nhau bằng việc cùng thực hiện một chính sách: yêu cầu các router trên vùng biên (diện tiếp xúc giữa các mạng con) giám sát các gói tin đi từ trong mạng nội hạt ra phía ngoài (các mạng con khác), và lọc bỏ nhũng gói tin có vấn đề, tức là các gói tin không có địa chủ nguồn xuất phát từ mạng nội hạt. Nhớ rằng những gói tin tấn công bao giờ cũng mang địa chỉ nguồn giả mạo, thường là ngẫu nhiên (không đời nào kẻ địch để lộ vị trí của mình, dù là địa chỉ mạng con của nó). Chính sách này có tên là *Ingress Filtering*, mặc dù trên lý thuyết sẽ rất lợi hại, tuy nhiên rất khó cài đặt thành công trên thực tế do khó có khả năng thực hiện được việc phối hợp liên hiệp của toàn bộ router của Internet.
* **Tổ chức cài đặt firewall (bức tường lửa) trên đường các gói tin đến với máy chủ.** Khi SYN đến firewall sẽ nhận rồi chuyển tiếp cho server, sau đó sẽ tạo và gửi ACK giả cho server để hoàn thành bắt tay, nhưng nếu sau khi đợi đủ lâu để khẳng định không có ACK thật sự từ Client tới, firewall sẽ gửi gói RST đến server đến đóng kết nối và giải phóng tài nguyên.
* **Chủ động giám sát.** Giám sát giao thông của TCP trong mạng cục bộ và tìm cách phát hiện các kết nối trái phép, do đó có thể gửi các gói RST để đóng các kết nối trái phép này.

## 8.4. Tấn công vào điều khiển tắc nghẽn TCP

Ở đây ta sẽ thảo luận về cơ chế điều khiển chống tắc nghẽn của TCP và một hình thức tấn công có thể xảy ra. Mục đích của việc điều khiển chống tắc nghẽn là đảm bảo để máy nguồn gửi tin không phát tin với tốc độ quá lớn, làm chật giải thông (bandwidth), dẫn tới tắc nghẽn cục bộ. Máy gửi cần phải thăm dò để tìm tốc độ phát tin thích hợp. Nó sẽ bắt đầu phát với một tốc độ thấp và tăng dần lên, trong khi đó theo dõi kích thước cửa sổ chưa ghi nhận dựa vào các gói tin phản hồi ACK nhận về. Khi kích thước cửa sổ chưa ghi nhận (số gói tin đã phát đi mà chưa được phản hồi tốt) vượt một ngưỡng đặt ra nào đó, tốc độ gửi tin sẽ phải điều chỉnh thấp đi.

Nguyên tắc của việc dò tìm tốc độ thích hợp này là *tăng cộng giảm nhân* (Additive Increase Multiplicative Decrease - AIMD), ngụ ý là việc tăng dần tốc độ (với xuất phát thấp) sẽ theo cấp số cộng, còn khi đã phát hiện điểm bão hòa thì cần giảm thật nhanh, tức là giảm theo cấp số nhân (chẳng hạn giảm theo kiểu chia đôi liên tục). Có giảm nhanh như vậy mới có khả năng tránh được tắc ngẽn. Khi đảm bảo kiểm soát được tốc độ dưới mức gây tắc ngẽn, hệ thống có thể lặp lại việc tăng dần tốc độ để tìm tốc độ tối ưu.

Một chiến thuật tấn công lợi hại có thể khai thác điểm yếu của cơ chế điều khiển trên như sau. Kẻ địch có mục đích phong tỏa giao thông khiến cho máy nạn nhân không thể liên lạc, truyền gửi tin với bên ngoài một cách thông suốt. Trong giai đoạn tăng dần tốc độ, kẻ địch sẽ đột ngột tạo một luồng phát tin mạnh cùng đồng thời với thời điểm máy nạn nhân đang gần ngưỡng tối ưu. Cơ chế điều khiển tránh nghẽn sẽ bắt buộc máy nạn nhân phải giảm sâu tốc độ đột ngột, thậm chí sau đó phải phát lại nhiều do mất mát vì tắc ngẽn. Kẻ địch cũng đột ngột “im lặng” để máy nạn nhân lại dần tăng tốc độ đến ngưỡng tối ưu, và sau đó lại lặp lại kịch bản đột ngột phát mạng để tạo xung đột tắc nghẽn. Như vậy kênh liên lạc của máy nạn nhân sẽ liên tục bị đứt quãng, thiếu ổn định, làm việc nhiều mà hiệu quả kém (phát lại nhiều, đến đích ít).

## 8.5 Bảo mật truyền tin tầng ip: giải pháp ipsec

IPSEC là một bộ giao thức phục vụ cho an ninh tầng IP thông qua cơ chế tác động lên các gói tin tầng IP để đảm bảo 3 mục tiêu: 1) Xác thực và toàn vẹn của thông tin; 2) Bảo mật và 3) Bảo vệ chống lại tấn công phát lại. Cơ chế cài đặt ở tầng IP làm cho việc sử dụng họ giao thức này trong suốt đối với tầng ứng dụng. Đây là một giải pháp tổng quát chung cho cộng đồng sử dụng Internet, được xây dựng bởi nhóm làm việc chuyên trách (IETF IPSEC Working Group).

IPSEC là bộ ba giao thức chính sau, cung cấp những dịch vụ thành phần

* Giao thức trao chuyển khóa IKE (Internet key exchange): chịu trách nhiệm khởi tạo cái gọi là liên kết an toàn (security association - SA), tức là một nhóm các thông tin điều khiển và tham số để sử dụng cho các thuật toán an toàn bảo mật cho cho liên kết, trong đó có các khóa sử dụng cho thuât toán mật mã và xác thực.
* Giao thức xác thức AH (Authentication Header): chỉ cung cấp cơ chế xác thực và bảo vệ tính toàn vẹn của gói tin, không đảm bảo tính bảo mật
* Giao thức đóng gói an toàn ESP (Encapsulating Security Payload): có 2 mức, mức cơ bản chỉ cung cấp dịch vụ bảo mật và mức nâng cao cung cấp toàn bộ tính bảo mật, xác thực và nguyên vẹn (tức là bao gồm cả các chức năng của AH).

Việc chia thành các thành phần dịch vụ tạo cơ hội cho khách hàng sử dụng có sử lựa chọn mềm dẻo, không bắt buộc phải sử dụng đầy đủ các tính năng vì chi phí có thể cao hơn và xử lý thông tin chậm hơn.

Cả hai giao thức AH và ESP này có thể hoạt động trong hai chế độ khác nhau:

* Chế độ giao vận (transport mode): dữ liệu từ tầng trên (TCP/UDP) được “bao bọc” theo một nghĩa nào đó (để đảm bảo xác thực và/hoặc bí mật) nhưng khối điều khiển IP header thì vẫn để nguyên. Cụ thể với ESP, toàn bộ dữ liệu truyền tải (IP payload) ngoại trừ IP header sẽ được mật mã và có thể được xác thực (tùy vào mức lựa chọn). Còn với AH, thì dữ liệu truyền tải và một phần được lựa chọn của IP header sẽ được xác thực
* Chế độ “đường hầm” (tunnel mode): toàn bộ dữ liệu, kể cả IP header, được bao bọc lại và một IP header mới được chèn thêm vào để chuyển tiếp trên mỗi chặng (giữa 2 router cùng hệ thống được cài IPSEC). Như vậy với ESP, toàn bộ gói tin IP gốc (kể cả IP header) sẽ được mật mã và có thể được xác thực. Còn với AH, việc xác thực được thực hiện trên toàn bộ gói tin IP gốc và một phần được lựa chọn của IP header mới thêm vào để chuyển tiếp trên mỗi chặng.

IPSEC có thể được sử dụng để bảo vệ các đường truyền dữ liệu giữa một cặp 2 máy (địa chỉ IP) tức là host-to-host, hoặc là giữa một cặp cổng an toàn (security gateways) tức là network-to-network, hoặc giữa một cổng và một máy tức là networkto-host.

## 8.6 Mối liên kết an toàn (security association)

Mối liên kết an toàn, gọi tắt là SA (security association), là cơ sở để xây dựng các chức năng an toàn bảo mật cho tầng IP. SA chẳng qua là một tập các lựa chon các giao thức mật mã cơ bản (về mật mã khóa đối xứng, mật mã khóa công khai, hàm băm, chữ ký điện tử …) và các tham số của chúng, nhằm đảm bảo cho các xử lý về mật mã và xác thực cho dòng thông tin truyền trên một chiều xác định (giữa hai máy xác định). Một máy tính chủ (host) có thể có cùng lúc nhiều SA để liên kết với nhiều máy khác, vì thế để xác định một SA duy nhất thì cần có chỉ số SPI (Security Parameter Index) và địa chỉ IP của máy đích của liên kết này.

Thông thường trong một đường trao đổi hai chiều giữa hai máy, các dòng truyền tin được đảm bảo an toàn bởi một cặp SA.

## 8.7 Giao thức AH (Authentication Header)

Giao thức này cung cấp khả năng đảm bảo tính toàn vẹn và xác thực cho các gói tin IP, dựa trên mã kiểm tra xác thực (MAC), cụ thể là lược đồ HMAC với các hệ hàm băm MD5 và SHA1. AH cũng đưa ra cơ chế chống tấn công phát lại (số thứ tự).

Cấu trúc cụ thể của phần header điểu khiển trong AH được mô tả qua hình 8.4, trong đó có các trường mang chỉ số mối liên kết an toàn (SPI) và số thứ tự (sequence number) như đã nói. Đặc biệt mã kiểm tra thông điệp sẽ chiếm 6 byte của trường 32-1. Khi Authentication Data. Trường số thứ tự chiếm 32 bit, nên có giá trị tối đa là 2 một SA mới được khởi tạo, trường số thứ này có giá trị 0, và sau đó các gói tin sẽ sử 32-1 được sử dụng thì liên kết SA sẽ dụng các giá trị tiếp theo. Nếu số thứ tự tối đa 2 phải khởi tạo lại, tức là các thông số điều khiển cần thượng lượng lại và các khóa mới được tạo ra. Số thứ tự lại được đưa về giá trị 0. Cơ chế đảm bảo cho việc số thứ tự không thể lặp lại trên cùng một SA và qua đó phát hiện được tấn công phát lại.

Hình 8.5 mô tả cấu trúc cụ thể của gói tin trong các chế độ giao vận và đường hầm. Qua hình vẽ có thể thấy rõ ràng trong chế độ giao vận, phần điều khiển (header) chỉ tác động lên thông tin từ tầng TCP chuyển xuống, bao gồm dữ liệu ứng dụng và điều khiển của TCP, trong khi đó trong chế độ đường hầm, phần điều khiển sẽ tác động lên của phần header của gói tin gốc. Ngoài ra trong chế độ đường hầm, vì toàn bộ gói tin gốc đã bị bao bọc (coi như một hàng hóa thuần để chuyển gửi), một header mới sẽ được sinh ra chèn thêm vào để dùng cho việc chuyển tiếp giữa các trạm trung gian, tức là các router trên đường truyền, có cài đặt bộ IPSEC này.

## 8.8 Giao thức đóng gói an toàn ESP

Như đã biết giao thức ESP cung cấp dịch vụ cơ bản là bảo mật và nếu lựa chọn, sẽ có thể cung cấp cả tính năng xác thực và toàn vẹn dữ liệu. Cấu trúc của gói tin được thể hiện trong hình vẽ 8.6. Phần thông tin được xác thực bao chứa phần thông tin bảo mật (không chứ phần chỉ số SPI và số thứ tự gói tin). Như với AH, trường Authentication Data chứa mã kiểm tra xác thực.

## 8.9 Bảo mật tầng tcp: họ giao thức ssl/tls

Bộ giao thức TLS (Transport Layer Security) là một giải pháp bảo mật phổ biến hàng đầu trên Internet. Ban đầu nó có tên là SSL (Secure Socket Layer), được khai sinh vào khoảng đầu thập kỷ 90 và phát triển như một sản phẩm của hãng Netscape Communications sau đó hãng này từ bỏ quyền sở hưu và đóng góp cho cộng đồng chung. Từ năm 1996, sản phẩm này được ủy nhiệm tiếp tục phát triển bởi một nhóm làm việc chuyên trách của IETF (Internet Engineering Task Force), cơ quan quốc tế phi chính phủ phụ trách về khởi thảo các chuẩn chung cho Internet.

TLS 1.0 ra đời chính thức vào năm 1999. Mặc dù đổi tên, TLS 1.0 không khác bao nhiêu so với SSL 3.0. Sự khác biệt này thậm chí còn nhỏ hơn so với sự khác biệt của SSL 2.0 và 3.0.

Như tên gọi ban đầu, giải pháp này nhằm hướng đến việc thiết lập những kênh giao tiếp an toàn và bảo mật giữa 2 tiến trình (process trên 2 máy). Bản thân khái niệm socket được đặt ra để mô tả một mối liên kết phiên làm việc giữa 2 tiến trình trên 2 máy (thông qua 2 cổng xác định, tức là TCP port). Về cơ bản, bộ giao thức cung cấp các tính năng như sau:

* Bảo mật: sử dụng các thuật toán mật mã đối xứng phổ biến DES, 3DES, RC2, RC4, IDEA
* Toàn vẹn dữ liệu: Sử dụng mã kiểm tra xác thực (MAC) với các hàm băm MD5, SHA1
* Trao chuyển khóa: sử dụng thuật toán khóa công khai (Diffie-Hellman và nâng cao)

## 8.10 Kiến trúc và các khái niệm cơ bản

TLS có thủ tục bắt tay chặt chẽ, cung cấp khả năng thương lượng giữa 2 bên để lựa chọn thuật toán cụ thể cho các giao dịch thiết lập khóa, truyền tin bảo mật và xác thực. Khác với IPSEC, cơ chế làm việc của TLS/SSL dựa trên giả thiết kênh truyền đã được đảm bảo liên lạc tin cậy (có thể coi như một dòng chảy liên tục của các gói tin, đúng thứ tự), nên nó được xây dựng ngay phía trên tầng TCP. Qua đó nó cung cấp dịch vụ an toàn bảo mật cho các giao thức làm việc ở tầng phiên và ứng dụng mà điển hình là HTTPS (kêt nối trình duyệt an toàn theo chuẩn HTML). SSL ban đầu đã là một sản phẩm mã nguồn mở và có nhiều bộ cài đặt khác nhau, điển hình là bộ SSLeay đã ra đời từ 2 thập kỷ trước

Về mặt kiến trúc, TLS/SSL thường được mô tả dưới dạng một chồng xếp của các giao thức (protocol stack), nằm ngay phía trên TCP (trong mô hình tham chiếu TCP/IP). Bản thân các giao thức con của nó được tổ chức thành 2 lớp con (sublayer). Bên tầng con dưới là giao thức SSL Record Protocol, coi như một “engine” (guồng máy làm việc), với nhiệm vụ xử lý mã hóa tất cả các thông tin dữ liệu từ bên trên giao xuống. Bên trên nó chính là các giao thức con làm nhiệm vụ quản lý điều khiển và “ngoại giao thương lượng” (SSL handshake/Change CipherSpec/Alert protocols) và các giao thức khai thác ứng dụng như HTTPS. Hình vẽ 8.8 là một thể hiện hình ảnh cách bố trí này.

Gắn liền với sơ đồ kiến trúc trên là hai khái niệm cơ bản: *phiên* (session) và *kết nối* (connection). Phiên được hình thành trên cơ sở một liên kết giữa hai máy, thường được xem như một client và một server, trong một khoảng thời gian nào đó. Các lựa chọn về thuật toán và thông số được thương lượng và xác định thông qua thủ tục bắt tay tạo phiên, tức là giao thức HandShake protocol. Thông qua thủ tục này, các thông số phiên làm việc sẽ được sử dụng chung bởi nhiều kết nối gắn trên phiên làm việc này. Trong khi đó, kết nối là một kênh truyền tin an toàn cụ thể, thường được gắn với hình ảnh socket, tức là một kết nối giữa 2 tiến trình cụ thể trên 2 cổng cụ thể giữa 2 máy client và server đã xác định. Tất nhiên mỗi kết nối được xác định trên một phiên đã thiết lập sẵn.

Ta hãy liệt kê một số những thuộc tính quan trọng của một phiên. Mã phiên (Session Identifier) được tạo bởi server để đinh danh một phiên tích cực hay có thể tái khởi động (resumable). Chứng chỉ khóa công khai (peer certificate) được tạo theo chuẩn X509. Phương pháp nén dữ liệu (compression method) để xác định thuật toán nén được chọn. Đặc tả mã mật (Cipher Spec) được dùng để chỉ định các thuật toán mật mã đối xứng và hàm băm (bao gồm thông số kích thước). Bản mật chính (Master secret) là một chuỗi 48 byte bí mật chung giữa hai bên client và server, mà dựa vào nó người ta tạo ra các khóa đối xứng chức năng cho các kết nối cụ thể sẽ được tạo ra theo phiên này. Có thể tái khởi động (Is resumable) là một thuộc tính logic cho phép hoặc không việc tái khởi động phiên làm việc để sinh ra những kết nối mới.

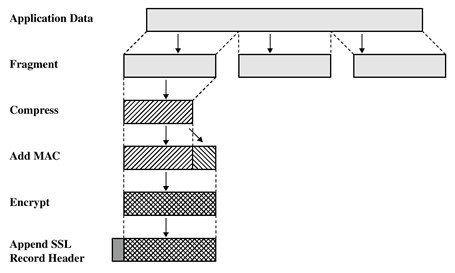
Các thuộc tính riêng của một kết nối cũng khá phong phú bao gồm nhiều khóa đối xứng chức năng (tạo trên cơ sở bản mật chính nói trên), đặc biệt được tạo ra thành nhiều cặp, tức là 2 khóa cùng chức năng nhưng phân biệt cho từng phía. Cụ thể là:

* Biến ngẫu nhiên của Client hay Server (Client/Server random): giá trị ngẫu nhiên được tạo ra để sử dụng vào các thao tác thách thức đáp ứng (cần cho thuật toán trao chuyển khóa)
* Giá trị mật tạo MAC phía server (Server write MAC secret): một khóa đối xứng chung mà server dùng để tạo mã kiểm tra xác thực còn client sử dụng để đối chứng.
* Giá trị mật tạo MAC phía client (Client write MAC secret): tương tự khóa trên nhưng cho client
* Khóa sinh mã phía server (Server write key): khóa đối xứng để server sinh mã còn client dùng để giải mã nhận được.
* Khóa sinh mã phía client (Client write key): tương tự trên nhưng cho client
* Vec-tơ khởi động (Initilization vecto): dùng cho các chế độ sinh mã, giá trị dùng chung cả 2 bên
* Số thứ tự gói tin (Sequence number): là con đếm được sử dụng bởi cả 2

64-1 (kết nối phải bên để ngăn chặn tấn công phát lại, đạt giá trị tối đa là 2 khởi động lại nếu đạt đến giá trị này).

## 8.11 Giao thức SSL Record protocol

Giao thức này chịu trách nhiệm xử lý thông tin mà tầng ứng dụng chuyển xuống, mật mã và đóng gói để chuyển tiếp xuống tầng IP. Các thao tác xử lý mà nó thực hiện là phân rã dữ liệu từ tầng ứng dụng thành các gói có kích thước phù hợp (fragment), nén dữ liệu để giảm kích thước rồi tạo ra mã xác thực để gửi kèm. Sau đó dữ liệu được mã hóa và chèn thêm khối thông tin điều khiển của tầng này trước khi gửi xuống tầng kế tiếp để truyền đi. Các chức năng xử lý này được minh họa rõ qua hình 8.9.



Hình vẽ 8.9: Quá trình xử lý dữ liệu của giao thức SSL Record protocol

Lưu ý rằng quá trình xử lý này sử dụng nhiều thuật toán (nén dữ liệu, tạo mã kiểm tra xác thực bằng hàm băm, sinh mã bằng thuật toán khóa đối xứng) mà các lựa chọn thuật toán cụ thể và thông số đã được xác định thông qua các giao thức của tầng con phía trên mà ta sẽ đề cập sau đây.

## 8.12 Giao thức bắt tay Handshake protocol

Như đã nói trên, giao thức này cung cấp cơ sở để hai bên bắt tay thương lượng và lựa chọn gói giải pháp cụ thể để truyền tin bảo mật (cipher spec), bao gồm:

* Thuật toán mật mã đối xứng
* Phương pháp xác lập khóa đối xứng (thuật toán trao chuyển khóa)
* Thuật toán hàm băm

Bên cạnh đó giao thức này cho phép hai bên thiết lập bản mật chính (master secret), để từ đó tạo ra các khóa chức năng cho từng kết nối mới sau nay. Giao thức cung cấp tùy chọn để hai bên client và server có thể tiến hành xác thực lẫn nhau (sử dụng chứng chỉ).

(Phần tô xám là các thông điệp tùy chọn)

Một cuộc giao dịch bắt tay có thể chia thành 4 pha như sau (hình 8.10). Trong pha thứ nhất, gồm 2 thông điệp chào hỏi và trao đổi thông tini là client\_hello và server\_hello, hai bên cũng nhau thương lượng các khả năng và chế độ an toàn, bao gồm các thông số như số phiên bản, mã phiên, đặc tả mật mã (các thuật toán mật mã), phương pháp nén và các giá trị ngẫu nhiên (nhãn thời gian và 28 byte ngẫu nhiên). Trong pha thứ hai, bao gồm tối đa 4 thông điệp gửi từ phía server, với ba thông điệp đầu là tùy chọn và 1 thông điệp cuối là tín hiệu kết thúc phần chào hỏi của server (server\_hello\_done). Trong 3 thông điệp tùy chọn nói trên, server có thể gửi sang chứng chỉ khóa của nó, thông tin cho thủ tục trao chuyển khóa và yêu cầu xem chứng chỉ của client. Tương ứng với các thông điệp tùy chọn nói trên, ở pha thứ ba, client sẽ trả lời bằng 3 thông điệp (cũng tùy chọn phù hợp với phía server) bao gồm việc chuyển chứng chỉ của client, thông tin cho thủ tục trao chuyển khóa và kết quả xác thực khóa server đã gửi. Pha 4 là pha kết thúc của cuộc bắt tay, nhưng bản thân nó cũng có thể tiến hành độc lập như một giao thức độc lập với tên gọi là *giao thức thay đổi đặc tả mật mã* (Change CipherSpec protocol). Client gửi cho server thông điệp chính thức để thiết lập đặc tả và thông số thuật toán mật mã và xử lý mới, đồng thời server sẽ gửi lại các thông tin đặc tả mà nó chấp nhận sử dụng. Tín hiệu kết thúc (finished) sẽ được gửi đi từ cả hai bên để kết thúc cuộc bắt tay và xác nhân các đặc tả thuật toán sẽ sử dụng.

Chú ý rằng cách tổ chức này cho phép *giao thức thay đổi đặc tả mật mã* (Change CipherSpec protocol) có thể thực hiện độc lập về sau này bất kỳ khi nào hai bên muốn để thiết lập lại các lựa chọn và thông số đặc tả mật mã. Điều này được khuyến khích làm để đảm bảo an toàn khi phiên làm việc đã thực hiện lâu (nhất là khi số thứ tự đã đạn ngưỡng tối đa cho phép).

Một phần quan trọng nằm lồng trong thủ tục bắt tay là nhiệm vụ trao đổi thông tin để xác lập khóa đối xứng (key exchange). Việc này thực hiện thông qua các thông điệp tùy chọn nói trên (tô xám trong hình vẽ). Có nhiệm phương án được cung cấp để tùy chọn:

* Sử dụng thuật toán RSA: khóa cần thiết lập được mã hóa bởi khóa công khai RSA
* Sử dụng thuật toán Diffie-Hellman, phương án lâu dài (Fixed DiffieHellman): các tham số công khai được cung cấp bằng chứng chỉ
* Sử dụng thuật toán Diffie-Hellman, phương án ngắn hạn (Ephemeral Diffie-

Hellman): các giá trị mật là ngắn hạn, các thông điệp ký bởi RSA hoặc DSS

* Sử dụng Diffie-Hellman, phương án ẩn danh (Anonymous Diffie-Hellman): không xác thực các khóa công khai; phương án này có thể bị tấn công bằng kiểu kẻ-ngồi-giữa (the man-in-the-middle)

## 8.13 PHÕNG VỆ CHO HỆ THỐNG KẾT NỐI MẠNG

Một hệ thống kết nối mạng, là một hệ thống có khả năng bị bộc lộ ra môi trường bên ngoài, có tiềm năng bị tấn công thông qua giao diện của nó với môi trường ngoài (mà phổ biến là Internet). Trước đây, các hệ thống thông tin của các tổ chức và doanh nghiệp lớn thường đóng kín do nhu cầu của việc bảo vệ các thông tin và dữ liệu quan trọng, nhạy cảm. Tuy nhiên sự phát triển lớn mạnh của Internet cùng với tính xã hội hóa cao của nó, hầu hết các hệ thống thông tin này đã phải ít nhiều mở ra kết nối với Internet, để có thể cung cấp dịch vụ và tự quảng cáo cho nó đến cộng đồng người sử dụng rất lớn trên Internet. Vì vậy đối với các chủ nhân của các hệ thống mở này, vấn đề đảm báo an toàn chống lại tấn công và khai thác trái phép thông qua Internet trở nên một vấn đề hết sức quan trọng, cấp bách. Các giải pháp về xác thực và điều khiển truy nhập mà ta nghiên cứu qua các chương trước đây chính là một trong những công cụ trọng yếu giúp bảo vệ các hệ thống này. Các giải pháp xác thực giúp xác minh đúng đối tượng người truy nhập hệ thống từ xa thông qua Internet trong khi điều khiển truy nhập sẽ cho chỉ cho phép người sử dụng khai thác hệ thống đúng theo phạm vi thẩm quyền của mình. Tuy nhiên vẫn còn rất nhiều vấn đề an toàn khác nằm ngoài giới hạn giải quyết của các giải pháp nói trên, trong đó có những vấn đề rất lớn như chống tấn công từ chối dịch vụ, hiện nay vẫn chưa có những giải pháp thực sự triệt để.

Trong khuôn khổ của một giáo trình cơ sở cho an toàn thông tin, chúng tôi không thể giới thiệu kỹ và rộng về một địa hạt lớn như thế này (an ninh mạng). Tiếp sau đây chúng tôi sẽ chỉ cung cấp thêm cho người đọc một số khái niệm cơ bản về vài giải pháp công cụ phổ biến nhất: *bức tưởng lửa, mạng riêng ảo* và *hệ dò tìm đột nhập*.

## 8.14 Bức tưởng lửa

*Bức tường lửa* (firewall, FW) là một hệ thống thiết bị và/hoặc phần mềm có chức năng chặn và “lọc” giao thông dữ liệu giữa hệ thống bên trong (cái cần phải bảo vệ) và môi trường bên ngoài (không đáng tin cậy). Thông thường, FW là một chương trình cài đặt trên một thiết bị chuyên dụng, bố trí như một cổng vào ra duy nhất cho liên lại giữa phía bên trong hệ thống cần bảo vệ và phía bên ngoài.

Mục đích chung của FW là nhằm giữ những “điều xấu” không thể lan vào bên trong hệ thống cần bảo vệ, vì vậy, các FW được cài đặt các chính sách an toàn thiết kế cụ thể để phòng tránh những “điều xấu” cụ thể có thể xảy ra này. Ví dụ như, có thể thiết kế chính sách để chỉ cho hệ thống phép liên lạc với (chấp nhận các packet đi tới/từ) một số địa chỉ hoặc người sử dụng trong một danh sách chỉ định, trong diện một số hoạt động cho phép (như các ứng dụng cụ thể nào đó). Về việc thiết kế và cài đặt các chính sách, có thể xảy ra tranh cãi, không thống nhất trong cộng đồng sử dụng (các NSD, nhà phát triển và chuyên gia an ninh). Cụ thể là có 2 xu hướng rõ rệt trong việc đề ra kiểu chiến lược ngầm định: “Cái gì không bị nêu rõ ràng là cấm thì có nghĩa là được phép” hoặc “Cái gì không nêu rõ ràng là được phép thì có nghĩa là bị cấm”. Giới NSD thường ưa thích kiểu ngầm định thứ nhất, trong khi giới quan trị và chuyên gia an ninh lại muốn áp dụng kiểu ngầm định thứ hai.

Có thể coi FW là một bộ máy giám sát (reference monitor), thường được cài đặt ở vị trí có thể điểu khiển, giám sát luồng giao thông vào/ra hệ thống. Nó cũng thường được cài đặt như một máy tính độc lập để khó bị tấn công đột nhập, đồng thời cũng được thiết kế gọn, tối giản để tiện lợi cho công tác phân tích và quản trị, bảo trì. Do tầm quan trọng như là cầu kết nối duy nhất này, hiệu năng xử lý cao (tốc độ) là điều rất quan trọng, đòi hỏi FW phải được tối ưu hóa về mặt chức năng, loại bỏ các chức năng phụ không thực sự liên quan đến nhiệm vụ chính. Thông thưởng mã chương trình FW được tối ưu hóa trên một hệ điều hành thu gọn, tối giản cho mục đích chính. Bên cạnh mục đích tăng cường hiệu năng, sự bố trí tối ưu cũng giúp cho việc phòng chống kẻ tấn công tiêm mã độc và chính chương trình của FW.

Sau đây ta sẽ điểm qua một số loại FW cơ bản và nguyên tắc hoạt động của chúng. Để tiện cho người đọc tra cứu tài liệu nước ngoài, chúng tôi để nguyên tên gọi tiếng Anh.

***Packet Filtering Gateway*** (PFG). Đây là FW loại đơn giản nhất nhưng cũng rất hiệu quả nếu dùng phù hợp: nó chỉ đơn giản là kiểm soát (lọc hay cho phép đi qua) các gói tin chỉ dựa vào thông tin ở header của chúng, cụ thể là địa chỉ IP nguồn và đích, địa chỉ TCP port nguồn và đích. Tất nhiên PFG không quan tâm đến nội dung bên trong của gói tin, tức là bản chất của dịch vụ mà gói tin này thuộc về. Hình 8.11 cho ta ví dụ minh họa trong đó, mạng LAN được bảo vệ bởi một PFG mà sẽ cho phép liên lạc tới một mạng bên ngoài trong khi chỉ định cấm liên lạc tới một mạng khác. Vì nguyên tắc kiểm soát khá đơn giản này, loại FW này có thể bị qua mặt bởi kẻ tấn công có thể tạo các gói tin với địa chỉ giả mạo, chính là loại mà nó cho phép đi qua.

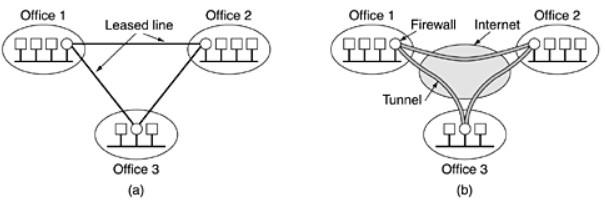
***Statefull Firewall*** (SF). Đây là loại FW “thông minh” hơn một chút, nó phần nào để ý đến trạng thái (state) của gói tin, tức là vai trò của gói tin trong một hoạt động của giao thức nào đó hay dịch vụ, tức là có thể theo dõi một chuỗi các gói tin trên một luồng, và qua đó phát hiện được những tấn công tinh vi hơn mà PFG đã bỏ qua.

***Application Proxy*** (AP). Còn được gọi là bastion host, là một loại FW phát triển phức tạp, có thể đóng vai trò đại diện, thay mặt một máy chủ ứng dụng (application server) để giao tiếp với các ứng dụng khách hàng (application client) bên môi trường ngoài. Thông qua việc kiểm soát các yêu cầu và hoạt động mà máy khách hàng tương tác với AP (máy khách hàng vẫn không biết mà “tưởng” là tương tác với máy chủ ứng dụng), quản trị hệ thống có thể cho phép những yêu cầu an toàn được chuyển tiếp tới máy chủ ứng dụng, hoặc loại bỏ những yêu cầu vượt quá phạm vi cho phép. Phần nào đó hoạt động kiểm soát của AP tương tự như hình thức hoạt động của kẻ tấn công trong kiểu tấn công kẻ-ngồi-giữa (the-man-in-the-middle attack).

## 8.15 Mạng riêng ảo

Với sự phát triển toàn cầu hóa nhanh chóng, thực tế cho thấy có rất nhiều công ty có nhiều chi nhánh, văn phòng hoặc nhà máy, nằm rải rác ở nhiều vùng, thành phố khác nhau, thậm chí là trên nhiều nước, nhiều châu lục khác nhau (các tập đoàn đa quốc gia). Trước đây, khi chưa phổ biến các mạng công cộng xuyên châu lục như mạng Internet, các công ty này này phải tốn tiền để cài đặt kênh thuê bao riêng của các công ty điện thoại và truyền thông khác, để có thể kết nối liên lạc giữa mỗi cặp chi nhánh. Các mạng kết nối như vậy được gọi là các mạng riêng (private networks). Cho đến ngày nay vấn tồn tại những mạng như vậy, dù rằng sự ra đời của Internet và hạ tầng truyền thống hiện đại đã đem lại các giải pháp mới, kinh tế hơn rất nhiều. Đương nhiên, các mạng riêng có thể đảm bảo tính an toàn rất cao, vì kẻ đột nhập tấn công bắt buộc phải thực hiện tấn công vào hạ tầng vật lý (thay vì các tầng cao hơn, dễ dàng hơn); tuy nhiên, thuê bao riêng với giải thông cao là rất đắt tiền. Giải pháp hiện đại thay thế cho các mạng sử dụng thuê bao riêng hiện nay là xây dựng *các mạng riêng ảo*, dịch từ *virtual private networks* – VPN, tức ra tạo ra một mạng con, kết nối logic giữa các điểm cần thiết trên nền của các mạng công cộng (chẳng hạn sử dụng các dịch vụ của các nhà cung cấp ATM hay Internet). Tất nhiên là người ta mong muốn thiết kế làm sao để tạo được sự an toàn cao, chống xâm nhập bên ngoài, để đảm bảo tính riêng biệt của mạng con này.

Ban đầu các VPN được xây dựng thông qua khai thác các mạng truyền thông cáp quang theo tiêu chuẩn ATM, nhưng xu thế chung hiện nay đang là xây dựng VPN trực tiếp trên Internet, thông qua những kết nối “đưởng hầm” (tunnel) giữa các nút. Một thiết kế phổ biến là xây dựng cho mỗi nút mạng VPN (ứng với mỗi văn phòng hay chi nhánh của công ty) một bức tưởng lửa (FW) và xây dựng một “đưởng hầm” giữa hai FW của 2 nút. “Đưởng hầm” này có thể đảm bảo bằng việc sử dụng IPSec, tức là một SA (security association) sẽ được thiết kế cho mỗi cặp nút, để đảm bảo một kênh truyền bảo mật, xác thực và ngăn cản mọi quan sát, phân tích nhìn từ bên ngoài, tức là không kém mấy so với việc sử dụng thuê bao riêng.



Hình 8.12: (a) Mạng riêng sử dụng thuê bao riêng; (b) VPN trên Internet

Hình 8.12 đưa ra minh họa về mạng riêng dùng thuê bao và VPN trên Internet sử dụng firewall và đường hầm kết nối. Mỗi firewall sẽ vừa tạo ra sự cách biệt cho mỗi nút với mỗi mô trường ngoài, vừa cung cấp cơ chế kết nối đường hầm trên cơ sở sử dụng IPSEC (thông thường các gói bảo mật này đã hỗ trợ sẵn trong phần mềm FW). Các nút thông qua FW sẽ bắt tay với nhau để thương lượng tham số bảo mật và thiết lập các SA, qua đó giao thực đóng gói bảo mật ESP sẽ được sử dụng trong chế độ đường hầm. Tóm lại ta có thể tạo nên một VPN qua sự kết hợp khá tự nhiên của việc dùng các FW, IPsec dùng với ESP trong chế độ đường hầm; cơ chế này tạo ra một vỏ bọc cách biệt mạng riêng ảo này với thế giới bên ngoài. Cơ chế này cũng có ưu điểm là tạo ra sự trong suốt, vô hình đối với các trình ứng dụng; chỉ các FW phải quan tâm và quản lý các SA. Chỉ duy nhất người quản trị phải biết đến và quan tâm đến cơ chế này thông qua việc quản lý và cấu hình các FW

## 8.16 Hệ thống dò tìm đột nhập

Hầu hết các cơ chế an toàn mà chúng ta đã đề cập trước đây, như xác thực người dùng hay bức tường lửa, nhằm phòng chống ngăn cản những kẻ tấn công từ bên ngoài. Tuy nhiên có một tỷ lệ rất cao các cuộc tấn công phá hoại được ghi nhận là từ bên trong, có thể là do những NSD có dã tâm hoặc do kẻ tấn công bên ngoài đã đột nhập thành công qua những bức tường bảo vệ nói trên. Việc dò tìm và phát hiện những tấn công loại này được thực hiện bởi các hệ thống có tên gọi là *hệ dò tìm đột nhập*, dịch từ *Intrusion Detection Systems* – IDS. Các hệ IDS thường được cài đặt như một thiết bị độc lập, phổ biến nhất là một máy tính với phần mềm chuyên dụng, nhằm tiến hành giám sát để phát hiện các hoạt động đáng nghi ngờ, có khả năng gây nguy hiểm đang diễn ra trong hệ thống cần bảo vệ. Sự hoạt động của một hệ thống có phần tương tự như một máy cảm biến phát hiện sự cố cháy ở các tòa nhà. Cơ chế hoạt động cơ bản của nó được minh họa qua hình 8.13.

Các chức năng chính của một hệ IDS như sau:

* Theo dõi giám sát các hoạt động của NSD và của hệ thống
* Kiểm soát thanh tra (auditing) cấu hình hệ thống để phát hiện các lỗi cấu hình hay các điểm nhạy cảm (vulnerabilities)
* Đánh giá tính toàn vẹn của phần lõi chủ chốt của hệ thống và các tệp dữ liệu
* Nhận dạng các mẫu tấn công đã biết qua theo dõi các hoạt động hệ thống
* Phát hiện và định danh các hoạt động bất thường thông qua các phân tích thông kê
* Sửa các lỗi cấu hình sai
* Cài đặt và vận hành bẫy hoạt động để tóm bắt thông tin về kẻ xâm nhập

Có một số loại IDS cơ bản như sau. Hệ *dò tìm dựa trên đặc tính dấu vết* (Signature-based IDS) hoạt động dựa trên việc phân tích phát hiện các tấn công đột nhập đã có lịch sử hoạt động trước đây (có thể ở nơi khác) mà cách thức và cơ chế hoạt động của chúng đã được ghi nhận và phân tích để tóm được dấu hiệu đặc trưng (signature). Tất nhiên các hệ IDS loại này chỉ phát hiện được các tấn công mà sự tồn tại là đã biết và phân tích về chúng là đủ nhiều để có thể ghi nhận được dấu hiệu đặc trưng. Các dấu hiệu đặc trưng này thường được chắt lọc dựa trên những phân tích thông kê. Tất nhiên người ta cố gắng để tìm ra những dấu hiệu đặc trưng mang tính khái quát cao, có thể phát hiện được cả các tấn công ngay kể cả khi nó đã có nhưng biến dị ít nhiều khác đi so với nguyên gốc.

Để thoát khỏi việc dò tìm chỉ dựa trên sự so khớp với những dấu hiệu đã biết của các tấn công đã tồn tại, hệ *dò tìm dựa trên tri thức kinh nghiệm* (heuristic IDS) để ý đến những dấu hiệu hoạt động bất thường, bao gồm những hành vi lạ, rất không điển hình của người dùng. Thông thường với những tấn công đột nhập tinh vi, mỗi hành vi cụ thể của nó thường tỏ ra hợp lệ, không phải là một dấu hiệu đáng nghi đã biết nào, nhưng sau một chuỗi các sự chuyển dịch tinh vi của những hành vi có vẻ như hợp lệ, hệ thống có thể rơi vào trạng thái bị thao túng bởi kẻ đột nhập. Vì vậy, để có thể phát hiện sự bất thường thông qua việc có thể xâu chuỗi các hành động liên tục để phân tích, người ta đưa ra một công cụ gọi lại *máy suy diễn* (inference engine). Những máy suy diễn này có thể hoạt động theo một trong hai nguyên lý cơ bản, dựa-trạng thái (state-based) hay dựa-mô hình (model-based). Chúng ta sẽ không đi sâu vào bản chất của chúng ở đây, nhưng có thể hiệu chung là cả hai nguyên lý này đều nhằm đến việc phát hiện ra được dấu hiệu của chuỗi hành vi đáng nghi, mà nó có thể dẫn hệ thống đến một trạng thái thiếu an toàn.

Mỗi hệ IDS là một thiết bị mạng, hay một phần mềm chạy trên một máy tính nối mạng, do vậy đều có thể bị tấn công và có thể bị vô hiệu hóa (chẳng hạn như bởi một tấn công DoS bố trí riêng biệt) làm mất chức năng bảo vệ hệ thống lớn. Tình huống rất có thể là, một kẻ tấn công khi đã có bước đột nhập thành công đầu tiên sẽ ngay lập tức tìm cách “khóa miệng” luôn hệ IDS bảo vệ cho hệ thống lớn. Biện pháp thường xuyên được dùng để tránh khỏi các tình huống trên là cài đặt các IDS trong chế độ ẩn, tàng hình (stealth mode). Cụ thể là một hệ IDS sẽ có hai giao diện (interface) kết nối mạng với hệ thống lớn, trong đó một để theo dõi và giám sát hoạt động hệ thống lớn và một là để có thể thông qua nó mà chủ động đưa ra báo động hoặc tín hiệu cảnh báo đến hệ thống lớn. Vì hoạt động theo nguyên lý “bị động”, tức là chỉ cho luồng gói tin đi qua và ghi nhận chứ không hề tự phát đi, giao diện mạng thứ nhất có thể được bố trí như tàng hình: địa chỉ của nó không hề được công bố ở đâu, đến các router bên trong hệ thống cũng không được tiết lộ. Khi cần chủ động đưa ra cảnh báo, hệ IDS sẽ sử dụng giao diện thứ hai, nằm hoàn toàn tách biệt trên một mạng khác.

Có một số hệ IDS đặc biệt được thiết kế với những nguyên lý riêng biệt. Phần mềm *tripwire* (1998) được biết tới như một công cụ phát hiện việc can thiệp sửa chữa mã chương trình trái phép. Ban đầu khi khởi động, nó tính toán giá trị băm của hàng loạt các tệp và dữ liệu hệ thống quan trọng rồi lưu trữ các giá trị băm này tại một nơi tách biệt an toàn. Sau này, khi cần kiểm tra xem hệ thống có bị xâm nhập và sửa đổi trái phép, các giá trị băm này có thể được tính lại và so sánh với các giá trị gốc đã lưu trữ, từ đó có thể phát hiện ra các nghi vấn tấn công. Một số hệ IDS cũng được thiết kế như các máy quét, ví dụ như *ISS Scanner*  hay *Nessus*, có thể dùng để kiểm tra và phát hiện các nghi vấn và lỗ hổng an toàn. Cuối cùng, các *giếng mật* (honeypot) cũng được coi như là các IDS, mà ở đó người ta thiết lập một môi trường giả lập với điều kiện dễ dàng và hấp dẫn để lôi kéo các tấn công đột nhập, qua đó bí mật quan sát và thu thập thông tin hành vi của các kẻ đột nhập này.