PHƯƠNG PHÁP BẢO MẬT CƠ SỞ DỮ LIỆU

Trong chiến lược bảo mật dữ liệu, đa số các công ty hiện nay tập trung nguồn lực vào bảo vệ dữ liệu trên đường truyền. Trong khi đó vấn đề bảo vệ dữ liệu nằm trong cơ sở dữ liệu (CSDL, database) chưa được quan tâm đúng mức. Thực tế cho thấy, sự cố về an ninh xảy ra với CSDL có thể ảnh hưởng nghi&

Trong chiến lược bảo mật dữ liệu, đa số các công ty hiện nay tập trung nguồn lực vào bảo vệ dữ liệu trên đường truyền. Trong khi đó vấn đề bảo vệ dữ liệu nằm trong cơ sở dữ liệu (CSDL, database) chưa được quan tâm đúng mức. Thực tế cho thấy, sự cố về an ninh xảy ra với CSDL có thể ảnh hưởng nghiêm trọng đến danh tiếng của công ty và quan hệ với khách hàng. Sự cố an ninh mất cắp 40 triệu thẻ tín dụng của khách hàng gần đây xảy ra với Master Card và Visa Card đã phần nào gia tăng sự chú ý đến các giải pháp bảo mật CSDL. Trong phạm vi bài này, người viết muốn trình bày các giải pháp bảo mật CSDL bằng phương pháp xây dựng tầng mã hóa. Giải pháp đơn giản nhất bảo vệ dữ liệu trong CSDL ở mức độ tập tin, chống lại sự truy cập trái phép vào các tập tin CSDL là hình thức mã hóa. Tuy nhiên, mã hóa dữ liệu ở mức độ này là giải pháp mang tính "được ăn cả, ngã về không", giải pháp này không cung cấp mức độ bảo mật truy cập đến CSDL ở mức độ bảng (table), cột (column) và dòng (row). Một điểm yếu nữa của giải pháp này là bất cứ ai với quyền truy xuất CSDL đều có thể truy cập vào tất cả dữ liệu trong CSDL. Điều này phát sinh một nguy cơ nghiêm trọng, cho phép các đối tượng với quyền quản trị (admin) truy cập tất cả các dữ liệu nhạy cảm. Thêm vào đó, giải pháp này bị hạn chế vì không cho phép phân quyền khác nhau cho người sử dụng CSDL. Giải pháp thứ hai, đối nghịch với giải pháp mã hóa cấp tập tin nêu trên, giải quyết vấn đề mã hóa ở mức ứng dụng. Giải pháp này xử lý mã hóa dữ liệu trước khi truyền dữ liệu vào CSDL. Những vấn đề về quản lý khóa và quyền truy cập được hỗ trợ bởi ứng dụng. Truy vấn dữ liệu đến CSDL sẽ trả kết quả dữ liệu ở dạng mã hóa và dữ liệu này sẽ được giải mã bởi ứng dụng. Giải pháp này giải quyết được vấn đề phân tách quyền an ninh và hỗ trợ các chính sách an ninh dựa trên vai trò (Role Based Access Control – RBAC). Tuy nhiên, xử lý mã hóa trên tầng ứng dụng đòi hỏi sự thay đổi toàn diện kiến trúc của ứng dụng, thậm chí đòi hỏi ứng dụng phải được viết lại. Đây là một vấn đề đáng kể cho các công ty có nhiều ứng dụng chạy trên nhiều nền CSDL khác nhau. Từ những phân tích hai giải pháp nêu trên, có thể dễ dàng nhận thấy một giải pháp bảo mật CSDL tối ưu cần hỗ trợ các yếu tố chính sau: 1. Hỗ trợ mã hóa tại các mức dữ liệu cấp bảng, cột, hàng. 2. Hỗ trợ chính sách an ninh phân quyền truy cập đến mức dữ liệu cột, hỗ trợ RBAC. 3. Cơ chế mã hóa không ảnh hưởng đến các ứng dụng hiện tại. Dưới đây là hai mô hình thỏa mãn các yêu cầu trên, đặc biệt là yêu cầu thứ ba. 1. Xây dựng tầng CSDL trung gian Trong mô hình này, một CSDL trung gian (proxy) được xây dựng giữa ứng dụng và CSDL gốc (Sơ đồ 1). CSDL trung gian này có vai trò mã hóa dữ liệu trước khi cập nhật vào CSDL gốc, đồng thời giải mã dữ liệu trước khi cung cấp cho ứng dụng. CSDL trung gian đồng thời cung cấp thêm các chức năng quản lý khóa, xác thực người dùng và cấp phép truy cập. Giải pháp này cho phép tạo thêm nhiều chức năng về bảo mật cho CSDL. Tuy nhiên, mô hình CSDL trung gian đòi hỏi xây dựng một ứng dụng CSDL tái tạo tất cả các chức năng của CSDL gốc. Hiện tại, trên thị trường sản phẩm mã hóa CSDL, Secure.Data của công ty Protegrity (www.Protegrity.com) sử dụng mô hình proxy nêu trên. 2. Sử dụng cơ chế sẵn có trong CSDL Mô hình này giải quyết các vấn đề mã hóa cột dựa trên các cơ chế sau: a. Các hàm Stored Procedure trong CSDL cho chức năng mã hóa và giải mã b. Sử dụng cơ chế View trong CSDL tạo các bảng ảo, thay thế các bảng thật đã được mã hóa. c. Cơ chế "instead of" trigger được sử dụng nhằm tự

động hóa quá trình mã hóa từ View đến bảng gốc. Trong mô hình này, dữ liệu trong các bảng gốc sẽ được mã hóa, tên của bảng gốc được thay đổi. Một bảng ảo (View) được tạo ra mang tên của bảng gốc, ứng dụng sẽ truy cập đến bảng ảo này. Truy xuất dữ liệu trong mô hình này có thể được tóm tắt như sau (Sơ đồ 2): Các truy xuất dữ liệu đến bảng gốc sẽ được thay thế bằng truy xuất đến bảng ảo. Bảng ảo được tạo ra để mô phỏng dữ liệu trong bảng gốc. Khi thực thi lệnh "select", dữ liệu sẽ được giải mã cho bảng ảo từ bảng gốc (đã được mã hóa). Khi thực thi lệnh "Insert, Update", "instead of" trigger sẽ được thi hành và mã hóa dữ liệu xuống bảng gốc. Quản lý phân quyền truy cập đến các cột sẽ được quản lý ở các bảng ảo. Ngoài các quyền cơ bản do CSDL cung cấp, hai quyền truy cập mới được định nghĩa: 1. Người sử dụng chỉ được quyền đọc dữ liệu ở dạng mã hóa (ciphertext). Quyền này phù hợp với những đối tượng cần quản lý CSDL mà không cần đọc nội dung dữ liệu. 2. Người sử dụng được quyền đọc dữ liệu ở dạng giải mã (plaintext). Giải pháp nêu trên có lợi điểm đơn giản, dễ phát triển. Tuy nhiên, do các giới hạn về cơ chế view, trigger và cách thức quản trị dữ liệu, giải pháp này có những hạn chế sau: Những cột index không thể được mã hóa, do đó hạn chế các ứng dụng cần hỗ trợ index Dữ liệu mã hóa có kích thước lớn so với dữ liệu gốc. Sự chênh lệch này không đáng kể đối với các dữ liệu chữ (text), nhưng rất đáng kể đối với các dữ liệu số và dạng nhị phân. Ví dụ, dữ liệu số 1 byte sẽ bị tăng lên 2 byte sau khi mã hóa. Tốc độ truy cập CSDL giảm do quá trình thực thi tầng mã hóa Hiện nay, trên thị trường sản phẩm mã hóa CSDL, DBEncrypt (www.appsecinc.com) và nCypher (www.ncypher.com) phát triển theo mô hình trên. Phùng Hải - CISSP, Project Manager, Global Cybersoft Vietnam