Firewalls and Internet Security
Second Edition

Repelling the Wily Hacker

William R. Cheswick
Steven M. Bellovin
Aviel D. Rubin

DRAFT COVER
as of 12/02

eastman
bring to you

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Chương 15: Phát hiện xâm nhập

“Kẻ ngốc nói rằng: ‘Đừng để tất cả trứng vào cùng một giỏ’ - đây chỉ là cách nói. Hãy ‘phân tán tiền và sự chú ý của bạn’. Nhưng người khôn ngoan nói rằng: ‘Hãy để tất cả trứng vào cùng một giỏ và... trông chừng cái giỏ đó!’”

— Puddin' Head Wilson's Calendar

Điều quan trọng là phải bố trí người canh gác gần những thứ bạn muốn bảo vệ, và hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) giúp thực hiện chức năng này. Là một sản phẩm thương mại, các công cụ bảo mật này đã được quảng bá như một giải pháp tối ưu cho các hành vi xâm nhập mạng, và nhiều nhà quản lý IT đã tuyên bố rằng mạng của họ an toàn vì họ đã cài đặt tường lửa và IDS mới nhất. Những điều này có thể hữu ích, nhưng chúng còn xa mới đạt đến mức lý tưởng.

Có một số loại hệ thống phát hiện xâm nhập. IDS mạng (Network IDS - NIDS) giám sát lưu lượng mạng để tìm các dấu hiệu của sự xâm nhập. Một số hệ thống dựa trên máy chủ quét các tệp hoặc lưu lượng truy cập để tìm virus; một số khác phân tích các mẫu cuộc gọi hệ thống hoặc kiểm tra các tệp bị thay đổi.

IDS phải đối mặt với một số hạn chế cố hữu. Dương tính giả (False positive) xảy ra khi IDS kết luận sai rằng một cuộc xâm nhập đã xảy ra. Âm tính giả (False negative) là các xâm nhập thực sự mà IDS bỏ sót. Với hầu hết các hệ thống phát hiện xâm nhập, cả hai vấn đề này đều không thể tránh khỏi và xảy ra với tần suất cao đến mức làm giảm đáng kể giá trị của chúng. Thông thường, cần có sự can thiệp của con người để xác định vấn đề hoặc thiếu sót, và một số nguồn gây ra lỗi trong các gói tin bị sai có thể quá khó để khắc phục.

Cuối cùng, các hệ thống IDS mạng thường hoạt động bằng cách “ngửi” lưu lượng mạng và ghép các gói tin lại thành dòng dữ liệu. Làm điều này một cách hợp lý nghe có vẻ đơn giản. Hầu hết các chương trình dò tìm chỉ làm điều đó, nhưng một số bài báo (như của Ptacek và Newsham [1998], Paxson [1998], và đặc biệt là Handley và cộng sự [2001]) chỉ ra rằng công việc này gần như không thể thực hiện chính xác. Vấn đề là một chương trình dò tìm cần biết trạng thái của các ngăn xếp TCP/IP ở cả hai đầu của liên lạc, cộng với các đặc điểm riêng của các chi tiết triển khai. Ví dụ, giả sử hai gói tin đến chứa dữ liệu chồng chéo nhau.

Vấn đề dữ liệu trùng lặp có vẻ như không phổ biến, nhưng các chương trình như fragrouter [Song *et al.*, 1999] được thiết kế để tạo luồng TCP/IP không bình thường nhằm gây nhầm lẫn cho các hệ thống phát hiện xâm nhập. Fragrouter sử dụng các kịch bản viết bằng một ngôn ngữ nhỏ để định nghĩa các vấn đề mong muốn trên luồng gói tin. Dữ liệu gửi đi có thể bị biến dạng đến mức hệ thống giám sát không thể giải mã được.

Dữ liệu chồng lấn có thể xuất hiện khi các gói tin được tái hợp lại và có hai phiên bản dữ liệu trùng lặp cho một vùng. Hệ thống sẽ sử dụng phiên bản nào? Các tiêu chuẩn RFC không đề cập đến vấn đề này, và các cách triển khai có thể khác nhau. Nếu dữ liệu trong hai gói tin không khớp, phiên bản nào mà Hệ thống phát hiện xâm nhập mạng (NIDS) sẽ cho là đúng?

Vấn đề dữ liệu chồng lấn nghe có vẻ hiếm, nhưng nó có thể xảy ra trong thực tế. Ví dụ, các chương trình như fragrouter [Song et al., 1999] cố tình thay đổi luồng TCP/IP để gây nhầm lẫn cho các hệ thống giám sát. Fragrouter sử dụng các đoạn mã nhỏ nhằm tạo ra các lỗi cụ thể trong dữ liệu gói tin. Kết quả là luồng gói tin đi ra có thể bị bóp méo đến mức hệ thống giám sát không thể giải mã luồng dữ liệu.

Có bốn nơi mà các luồng TCP/IP bất thường cần được xử lý đúng cách: khách hàng, máy chủ, tường lửa, và NIDS. [Handley et al., 2001] đề xuất sử dụng một thiết bị trung gian để chuẩn hóa luồng gói tin. Một cách tiếp cận khác là bổ sung chức năng này vào tường lửa, làm cho nó hoạt động giống như một cổng mạch (circuit-level gateway). Một số tường lửa đã thực hiện điều này; chúng tái hợp các gói tin bị phân mảnh để bảo vệ khỏi các cuộc tấn công bằng gói tin ngắn. Các cổng mạch thực sự cũng có thể làm sạch luồng IP.

Vấn đề này là cơ sở cho khuyến nghị của chúng tôi trong lần xuất bản đầu tiên rằng các tổ chức nên tránh kết nối IP trực tiếp giữa mạng nội bộ và Internet, và thay vào đó sử dụng tường lửa ứng dụng hoặc tường lửa cấp độ mạch.

15.1 Nơi cần giám sát

Điều quan trọng là phải hiểu những hạn chế của Hệ thống Phát hiện Xâm nhập (IDS) trước khi triển khai. Một câu hỏi cần đặt ra là: "Mục đích của IDS là gì?" Một lý do chính đáng để cài đặt IDS ngoài tường lửa là để biện minh cho việc cấp ngân sách (vì đây là một mô hình mối đe dọa trong đó quản lý là "kẻ thù"). Không cần thiết phải giám sát lưu lượng ngoài mạng của bạn để xem liệu bạn có đang bị tấn công hay không — bạn biết mình là mục tiêu. Điều đó không có nghĩa là bạn nên bỏ qua lưu lượng bên ngoài, nhưng tốt hơn là ghi lại và lưu trữ lưu lượng bên ngoài để phân tích sau, thay vì cố gắng phát hiện xâm nhập trong thời gian thực.

Nếu bạn là nhà nghiên cứu học hỏi về các cuộc tấn công mới, thông tin như vậy là vô giá. Tuy nhiên, có quá nhiều lưu lượng xảy ra, và IDS là công cụ quá yếu để thực hiện phân tích thời gian thực. Đây là một công cụ tốt để huấn luyện những người mới làm quen với mạng và thiết bị IDS.

Thiết bị IDS trở nên hữu ích hơn khi được triển khai gần các tài sản quan trọng, bên trong các lớp bảo mật khác nhau. Chúng giống như một camera an ninh được lắp đặt trong két sắt của ngân hàng, cung cấp một lớp bảo đảm cuối cùng rằng mọi thứ đều ổn. Khi quyền truy cập thông thường vào mạng hoặc máy tính bị giới hạn nghiêm ngặt hơn, các quy tắc dành cho thiết bị phát hiện cũng cần phải nhạy cảm hơn. Con người không nên thực hiện các truy vấn web từ máy tính bảng lương.

15.2 Các loại IDS

Các loại IDS khác nhau có điểm mạnh và điểm yếu khác nhau.

IDS dựa trên chữ ký (Signature-based IDS):
Loại này có một cơ sở dữ liệu các cuộc tấn công đã biết; bất kỳ điều gì khớp với mục trong cơ sở dữ liệu sẽ bị gắn cờ. Bạn sẽ không gặp nhiều cảnh báo sai nếu hệ thống được điều chỉnh đúng cách, nhưng khả năng cao sẽ bỏ sót các cảnh báo đúng vì hệ thống chỉ nhận diện những gì nằm

trong cơ sở dữ liệu. Thật không may, việc tìm được sự cân bằng giữa chữ ký rộng (khớp với lưu lượng bình thường) và chữ ký hẹp (dễ bị mã độc qua mặt) là rất khó. Tốt nhất, các hệ thống dựa trên chữ ký nên tích hợp ngữ cảnh thay vì chỉ dựa vào việc so khớp chuỗi.

IDS dựa trên bất thường (Anomaly-based IDS):
Những hệ thống này tìm kiếm các hành vi không bình thường và có khả năng đưa ra các cảnh báo sai hoặc bỏ sót cảnh báo đúng. Chúng hoạt động tốt nhất trong môi trường với phiên bản định nghĩa chặt chẽ về "bình thường," nơi dễ xác định khi một điều gì đó không nên xảy ra. Mục đích càng đặc thù của một máy, thì hành vi "bình thường" càng bị giới hạn, và khả năng cảnh báo sai càng giảm.

Phát hiện bất thường là một lĩnh vực nghiên cứu thú vị, nhưng đến nay vẫn chưa tạo ra nhiều công cụ thực tiễn. [Forrest et al., 1996] và [Ko et al., 2000] đã cho thấy một số kết quả đáng chú ý. Forrest phát triển một công cụ giám sát các quy trình chạy trên máy tính và kiểm tra các lệnh hệ thống. Công cụ này có khái niệm về mẫu lệnh "bình thường" và nhận ra khi điều gì đó không đúng xảy ra. Nó sử dụng n-grams của các lệnh hệ thống và phân tích thứ tự thực thi để phát hiện bất thường.

IDS dựa trên máy chủ hoặc mạng (Host-based & Network-based IDS):
Hai loại này bổ sung cho nhau, không loại trừ nhau. Hệ thống dựa trên máy chủ thường biết trạng thái của chính máy đó, giúp xử lý dữ liệu dễ hơn, nhưng phần mềm có thể bị xâm phạm nếu máy chủ bị tấn công. Hệ thống dựa trên mạng là các thiết bị độc lập và thường ít bị tấn công hơn.

Một số môi trường như DMZ (vùng phi quân sự hóa) yêu cầu một loại IDS đặc biệt gọi là honeypot — một máy tính mà không ai được phép truy cập. Bất kỳ lưu lượng nào đến máy này có thể được xem là hành vi đáng ngờ. Honeypot không phù hợp trong môi trường sản xuất mở, nhưng nó phù hợp để phát hiện các cuộc tấn công mà không ảnh hưởng đến các tài nguyên dành riêng.

Một hệ thống honeypot (bẫy mật) trên mạng Internet công cộng có thể hữu ích để nghiên cứu hành vi của hacker, mặc dù một số hacker đã học cách né tránh chúng. Một trong những ví dụ đẹp nhất là *honeyd* của Niels Provos [Spitzner, 2002, Chương 8]. Nó mô phỏng một mạng lưới hoàn chỉnh, bao gồm nhiều loại máy móc khác nhau. Tuy nhiên, bạn không thể dựa vào nó để xác định liệu ai đó đã xâm nhập vào một máy đơn lẻ hay chưa; tối đa, nó chỉ có thể phát hiện các lượt quét. Để xử lý các cảnh báo dương tính giả và âm tính giả, một số người sử dụng nhiều hệ thống IDS (Hệ thống Phát hiện Xâm nhập) có đầu ra được đồng bộ hóa. Sự tương quan thời gian có thể được sử dụng để phát hiện các cuộc tấn công "chậm và âm thầm."

15.3 Quản trị một hệ thống IDS

Một hệ thống phát hiện xâm nhập (IDS) yêu cầu một lượng tài nguyên đáng kể. IDS phải được cài đặt ở các vị trí chiến lược, cấu hình đúng cách và giám sát thường xuyên. Trong hầu hết các môi trường, chúng sẽ phải xử lý một lượng lớn lưu lượng mạng không hợp lệ. Ví dụ, một trình điều khiển máy in HP mà chúng tôi đã sử dụng đã cố tìm mọi thứ trên mạng con mà không biết về mặt nạ mạng, dẫn đến quét toàn bộ mạng /16 để tìm một máy in HP. Phần mềm quản lý mạng

đôi khi cũng làm điều tương tự. Người vận hành IDS phải biết cách xử lý loại lưu lượng này và cần có khả năng chịu đựng một lượng lớn "tiếng ồn" [Bellovin, 1993]. Họ cũng cần đảm bảo rằng họ không trở nên quá tự mãn vì IDS có xu hướng "báo động giả."

15.4 Công cụ IDS

Có rất nhiều công cụ IDS có sẵn, cả miễn phí lẫn thương mại. Các công cụ như snort (xem phần sau), ethereal, và bro [Paxson, 1998] rất hữu ích. Ethereal cung cấp một giao diện đồ họa (GUI) đẹp, cho phép bạn tái hiện các luồng TCP để xem dữ liệu ở cấp độ ứng dụng. Nó cũng có thể ghi lại lưu lượng mạng để phân tích sau.

Các sản phẩm thương mại dao động từ kém chất lượng đến khá hữu ích. Một số sản phẩm cố gắng áp dụng kỹ thuật trí tuệ nhân tạo để giải quyết vấn đề. Một số khác thu thập thông tin phân tán và cố gắng lắp ráp một cái nhìn tổng thể về cuộc tấn công.

15.4.1 Snort

Có lẽ chương trình phát hiện xâm nhập miễn phí phổ biến nhất là snort, được phát triển bởi Martin Roesch. Snort là mã nguồn mở, và có một cộng đồng lớn các người dùng và người đóng góp tích cực. Xem thêm tại:
http://www.snort.org/
Chương trình có sẵn trên nhiều nền tảng — nó hoạt động ở bất kỳ đâu mà libpcap chạy được.

Snort có thể được sử dụng theo nhiều cách khác nhau. Nó có thể "ngửi" mạng và tạo đầu ra theo định dạng tcpdump. Nó cũng có thể được sử dụng để ghi lại các gói tin, để các công cụ khai thác dữ liệu hoặc chương trình của bên thứ ba phân tích lưu lượng mạng sau đó. Tính năng thú vị nhất của snort là khả năng thiết kế một bộ quy tắc nhận diện các mẫu lưu lượng cụ thể. Nhiều quy tắc đã có sẵn cho snort và chúng thường được chia sẻ giữa người dùng và đăng tải trên Internet. Snort có thể được cấu hình để nhận diện các khảo sát bằng nmap, các cuộc tấn công tràn bộ đệm đã biết, các khai thác CGI đã biết, và các lưu lượng thăm dò như quét cổng.

Những nỗ lực để nhận dạng hệ điều hành dựa trên các đặc điểm của ngăn xếp mạng, và nhiều kiểu tấn công khác mà quản trị viên muốn cấu hình quy tắc. Dưới đây là một ví dụ quy tắc được lấy từ [Roesch, 1999]:

```
bash
activate tcp !$HOME_NET any -> $HOME_EIET 143 (flags: PA; content:
"|E8C0FFFFFF|bin|; activates: 1; msg: "(buffer overflow!");
dynamic tcp !$HOME_NET any -> $HOME_NET 143 (activated_by: 1; count: 50;);
```

Quy tắc trên chỉ ra rằng một cảnh báo sẽ được gửi khi một lỗi tràn bộ đệm IMAP được phát hiện. Tại thời điểm đó, 50 gói tin đến tiếp theo hướng đến cổng 143 sẽ được ghi lại. Một số trong những gói tin này có thể chứa thông tin quan trọng về cuộc tấn công mà một nhà phân tích hoặc quản trị viên mạng quan tâm.

Tuy nhiên, có một lỗ hổng: Quy định về cờ "PA" có nghĩa là cả hai bit PUSH và ACK phải được đặt trên gói tin để nó phù hợp với quy tắc này. Điều này khá dễ để kẻ tấn công né tránh bằng cách đảm bảo rằng PUSH không được đặt.

Như mong đợi từ các công cụ phát hiện xâm nhập hữu ích, Snort cung cấp các cơ chế cảnh báo linh hoạt, từ cửa sổ bật lên trên màn hình đến email và thông báo qua thiết bị nhắn tin. Có các nhóm người dùng Snort thường tụ họp để so sánh dữ liệu, chia sẻ tập luật, thảo luận về kết quả dương tính giả, và đề xuất các cải tiến có thể cho chương trình. Ngoài ra, còn có một diễn đàn trực tuyến với rất nhiều thông tin hữu ích tại http://snort.rapidnet.com/.

Và đúng vậy, luôn có một "cuộc chạy đua vũ trang" giữa những kẻ tấn công và các nhà phát triển script Snort.