Thích ứng Trình phát hiện Bất thường Tự động bằng Adaptive Điều chỉnh ngưỡng

MUHAMMAD QASIM ALI và EHAB AL-SHAER, Đại học Bắc Carolina Charlotte (UNCC)
HASSAN KHAN, Đại học Khoa học và Công nghệ Quốc gia (NUST), Pakistan
SYED ALI KHAYAM, PLUMgrid Inc

Hệ thống phát hiện bất thường dựa trên máy chủ và mạng thời gian thực (ADS) chuyển đổi luồng dữ liệu đầu vào liên tục thành điểm số bất thường có ý nghĩa và có thể định lượng. Những điểm số này sau đó được so sánh với ngưỡng phát hiện cố định và được phân loại là lành tính hoặc độc hại. Chúng tôi lập luận rằng đầu vào của ADS thời gian thực thay đổi đáng kể theo thời gian và giá trị ngưỡng cố định không thể đảm bảo độ chính xác phát hiện bất thường tốt cho đầu vào thay đổi theo thời gian như vậy. Trong bài viết này, chúng tôi đề xuất một kỹ thuật chung và đơn giản để điều chỉnh thích ứng ngưỡng phát hiện của bất kỳ ADS nào hoạt động theo phương pháp ngưỡng. Để đạt được điều này, trước tiên, chúng tôi thực hiện phân tích thống kê và lý thuyết thông tin về điểm số bất thường của ADS dựa trên mạng và máy chủ để tiết lộ cấu trúc tương quan thời gian nhất quán trong các giai đoạn hoạt động lành tính. Chúng tôi lập mô hình cấu trúc quan hệ tương quan được quan sát bằng cách sử dụng chuỗi Markov, lần lượt được sử dụng trong khung theo dõi mục tiêu ngẫu nhiên để điều chỉnh ngưỡng phát hiện của ADS theo các phép đo thời gian thực. Chúng tôi cũng sử dụng các kỹ thuật thống kê để làm cho thuật toán được đề xuất có khả năng phục hồi trước các thay đổi lẻ tẻ và các cuộc tấn công trốn tránh. Để đánh giá phương pháp được đề xuất, chúng tôi kết hợp mô-đun ngưỡng thích ứng được đề xuất vào nhiều ADS và đánh giá các ADS đó trên bộ dữ liệu tấn công máy chủ và mạng được thu thập toàn diện và độc lập. Chúng tôi cho thấy rằng, trong khi giảm nhu cầu cấu hình ngưỡng của con người, kỹ thuật được đề xuất cung cấp các cải tiến về độ chính xác đáng kể và nhất quán cho tất cả các ADS được đánh giá.

Danh mục và Mô tả chủ đề: C.2.0 [Mạng máy tính-truyền thông]: Chung—An ninh và bảo vệ (ví dụ: tường lửa); D.4.6 [Hệ điều hành]: Bảo mật và Bảo vệ— Phần mềm xâm lấn; K.6.5 [Quản lý hệ thống máy tính và thông tin]: Bảo mật và bảo vệ—Trái phép truy cập

Thuật ngữ chung: Thuật toán, Bảo mật

Các từ và cụm từ khóa bổ sung: Ngưỡng thích ứng, phát hiện bất thường, phát hiện xâm nhập, bất thường điểm số

Định dạng tham chiếu ACM: Ali,
MQ, Al-Shaer, E., Khan, H. và Khayam, SA 2013. Tự động điều chỉnh trình phát hiện bất thường bằng cách sử dụng điều chỉnh ngưỡng thích ứng. ACM Trans. thông tin liên lạc hệ thống. bảo mật. 15, 4, Điều 17 (04/2013), 30 tr.
DOI: http://dx.doi.org/10.1145/2445566.2445569

1. GIỚI THIỆU

Những thập kỷ qua đã chứng kiến sự gia tăng chưa từng có về số lượng và mức độ phức tạp của các cuộc tấn công mạng. Sự gia tăng theo cấp số nhân của các cuộc tấn công zero-day đã

Một phần của công việc này đã xuất hiện trong Kỷ yếu của Hội nghị ACM về Bảo mật Máy tính và Truyền thông (CCS), 2009 [Ali et al. 2009].
Địa chỉ của các tác giả: MQ Ali (Tác giả tương ứng) và E. Al-Shaer, Khoa Phần mềm và Hệ thống Thông tin, Đại học North Carolina Charlotte, 9201 University City Blvd. Charlotte, NC 28223; email: mohdqasimali@gmail.com; H. Khan và SA Khayam, Trường Kỹ thuật Điện và Khoa học Máy tính, Đại học Khoa học và Công nghệ, Pakistan.

1094-9224/2013/04-ART17 $15,00 DOI:http://dx.doi.org/10.1145/2445566.2445569

nhấn mạnh sự cần thiết của các cơ chế phòng thủ có thể phát hiện chính xác các cuộc tấn công chưa từng thấy trước đây trong thời gian thực mà ít (nếu có) sự can thiệp của con người. Do đó, nỗ lực nghiên cứu quan trọng đã được nhắm mục tiêu vào phát hiện bất thường dựa trên mạng và máy chủ thời gian thực [Agosta et al. 2007; Forrest và cộng sự. 1996; Twycross và Williamson 2003]. Trái ngược với các hệ thống phát hiện xâm nhập dựa trên chữ ký truyền thống phát hiện các vectơ tấn công đã biết, Hệ thống phát hiện bất thường (ADS) phát hiện các sai lệch so với hành vi bình thường để phát hiện các cuộc tấn công zero-day.

Hầu hết các ADS thời gian thực đều yêu cầu sự can thiệp của con người (về mặt trực giác và kinh nghiệm) để đặt các ngưỡng phát hiện bất thường (bất biến về thời gian và hành vi), và do đó không có khả năng thích ứng với các đặc điểm lưu lượng truy cập và máy chủ lưu trữ khác nhau.
Để xác định ngưỡng hoạt động cho ADS, các đường cong Đặc tính hoạt động của máy thu (ROC) được tạo bằng cách áp dụng một loạt các ngưỡng phân loại cho điểm số bất thường của ADS, sau đó vẽ đồ thị tỷ lệ phát hiện so với tỷ lệ cảnh báo sai cho từng giá trị ngưỡng. Đối với hoạt động thời gian thực, điểm vận hành ROC tốt nhất được chọn để vượt ngưỡng điểm bất thường. Trong thực tế, ngưỡng cứng nhắc như vậy dẫn đến mất tự động hóa và độ chính xác [Ashfaq et al. 2008; Lippmann và cộng sự. 2000] và do đó cản trở việc triển khai rộng rãi các ADS [Báo cáo của Gartner]. Để tự động hóa quy trình này, một số sản phẩm ADS dựa trên mạng thương mại điều chỉnh ngưỡng của chúng theo các đặc điểm lưu lượng đầu vào [Arbor PeakFlow; Cisco Anomaly Guard] và một số ADS gần đây cung cấp các phương pháp để tính ngưỡng tối ưu cho các thuật toán cụ thể của họ [Ide và Kashima 2004; Jung et al. 2004; Lakhina và cộng sự. 2004].

Tuy nhiên, một kỹ thuật thích ứng ngưỡng chung , có thể được cắm vào ADS hiện có phần lớn chưa được khám phá.

Trong bài viết này, chúng tôi thách thức sự hiểu biết thông thường rằng các đường cong ROC được tạo bằng cách sử dụng các ngưỡng cố định có thể xác định độ chính xác tốt nhất có thể đạt được cho một ADS. Chúng tôi lập luận rằng các đặc điểm đầu vào của ADS thời gian thực thay đổi liên tục theo thời gian và việc đặt ngưỡng phân loại cứng nhắc (bất biến về thời gian và hành vi) giới hạn độ chính xác mà ADS có thể đạt được. Ngoài ra, việc xác định ngưỡng bằng cách sử dụng các đường cong ROC cũng đưa ra sự can thiệp không mong muốn của con người vào hoạt động của ADS. Để giảm thiểu những hạn chế này, chúng tôi đề xuất một kỹ thuật để điều chỉnh thích ứng ngưỡng phát hiện của ADS thời gian thực phù hợp với các hành vi mạng và máy chủ khác nhau.

Tiền đề cơ bản của kỹ thuật thích ứng ngưỡng được đề xuất là: Nếu chúng ta có thể dự đoán chính xác các giá trị dự kiến của điểm số bất thường trong tương lai dưới các điều kiện lành tính, thì ngưỡng phân loại có thể được điều chỉnh như một hàm của điểm số dự đoán.
Do đó, việc thích ứng ngưỡng yêu cầu một thuật toán theo dõi có thể dự đoán chính xác điểm số bất thường trong tương lai. Các hạn chế bổ sung đối với kỹ thuật thích ứng ngưỡng thực tế là: (1) nó có thể tự động tìm hiểu hành vi tạm thời của các điểm bất thường trong các điều kiện thuận lợi; (2) nó phải duy trì ổn định khi có những thay đổi ngắn hạn không thường xuyên ở đầu vào (bùng nổ lưu lượng truy cập); (3) nó sẽ cho phép ADS đạt được các điểm có độ chính xác tốt trên mặt phẳng ROC; (4) nó phải chung chung để có thể dễ dàng tích hợp vào ADS hiện tại hoạt động theo cách ngưỡng, bất kể các nguyên tắc phát hiện và tính năng máy chủ/lưu lượng truy cập được sử dụng bởi ADS; và (5) nó phải có độ phức tạp trong thời gian chạy thấp.

Theo quan điểm của những hạn chế này, trước tiên chúng tôi tiết lộ một số thuộc tính thống kê chung về điểm số bất thường của ADS. Cụ thể, chúng tôi sử dụng phân tích tự tương quan và phân tích en tropy có điều kiện để chỉ ra rằng điểm số bất thường theo thời gian thực trong các điều kiện lành tính thể hiện cấu trúc phụ thuộc theo thời gian đang phân rã. Do đó, điểm số trong tương lai có thể được dự đoán chính xác bằng cách sử dụng một vài điểm số trước đó. Vì một quy trình ngẫu nhiên với sự phụ thuộc theo thời gian giảm dần có thể được mô hình hóa chính xác bằng cách sử dụng chuỗi Markov, nên chúng tôi cho thấy rằng thuật toán theo dõi mục tiêu dựa trên Markov có độ phức tạp thấp có thể dự đoán chính xác điểm số bất thường trong tương lai. Để cung cấp khả năng phục hồi chống lại kẻ tấn công lảng tránh, chúng tôi sử dụng bộ lọc thông thấp để loại bỏ các đột biến ồn ào trong thời gian ngắn khỏi điểm số bất thường quan sát được

và bình thường hóa lỗi dự đoán theo độ lệch chuẩn của điểm bất thường trong một khoảng thời gian. Những cải tiến này cho phép công cụ dự đoán tránh tìm hiểu các thay đổi ngắn hạn lẻ tẻ trong hành vi đầu vào, đồng thời dần dần tìm hiểu các xu hướng điểm số bất thường trong dài hạn. Chúng tôi cho thấy rằng thuật toán Markovian được đề xuất cung cấp độ chính xác dự đoán tốt hơn đáng kể so với bộ lọc Kalman thông thường và bộ lọc trước Holt-Winters. Điểm bất thường được dự đoán bằng thuật toán đề xuất được sử dụng để xác định ngưỡng phân loại cho các phép đo thời gian thực tiếp theo.

Chúng tôi chứng minh các cải tiến về độ chính xác (tốc độ phát hiện và cảnh báo sai) có thể đạt được bằng cách sử dụng kỹ thuật được đề xuất bằng cách kết hợp mô-đun ngưỡng thích ứng vào tám Hệ thống Phát hiện Bất thường (ADS) nổi bật và đa dạng trên mạng và máy chủ. ADS thích ứng được đánh giá trên bảy bộ dữ liệu được dán nhãn và thu thập độc lập. Tất cả các bộ dữ liệu được sử dụng trong công việc này đã được cung cấp công khai ngoại trừ bộ dữ liệu mạng dựa trên tải trọng do thỏa thuận về quyền riêng tư và không tiết lộ. Chúng tôi cho thấy rằng các cải tiến đáng kể và nhất quán đối với các điểm vận hành ROC tốt nhất có thể đạt được bằng các ADS thích ứng. Chúng tôi cũng chỉ ra rằng kỹ thuật ngưỡng thích ứng được đề xuất có khả năng đối phó với kẻ tấn công lảng tránh, kẻ có thể cố gắng làm mất ổn định dự đoán bằng cách đưa ra những thay đổi dao động lẻ tẻ trong hành vi lưu lượng. Cuối cùng, chúng tôi chỉ ra rằng ngưỡng thích ứng có độ phức tạp không đáng kể so với độ phức tạp của thuật toán phát hiện bất thường ban đầu.

Phần còn lại của bài viết được tổ chức như sau. Phần 2 mô tả bối cảnh và công việc liên quan trong lĩnh vực này. Động lực và phương pháp cốt lõi của công việc đề xuất được thảo luận trong Phần 3. Phần 4 mô tả ngắn gọn các bộ dữ liệu và bộ phát hiện bất thường được sử dụng trong công việc này (chi tiết về bộ dữ liệu và bộ phát hiện có thể được tìm thấy trong Phụ lục A và B tương ứng). Trong Phần 5, chúng tôi thực hiện phân tích thống kê về điểm số của ADS.
Dựa trên phân tích thống kê, trong Phần 6, thuật toán ngưỡng thích ứng được đề xuất cùng với tính ổn định và khả năng trốn tránh mạnh mẽ của nó. Đánh giá độ chính xác và độ phức tạp của phương pháp đề xuất được cung cấp trong Phần 7. Cuối cùng, Phần 8 tóm tắt các kết luận chính của công việc này.

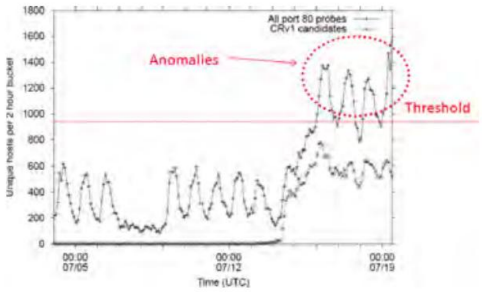
2. BỐI CẢNH VÀ CÁC CÔNG VIỆC LIÊN QUAN

Trong phần này, trước tiên chúng tôi cung cấp thông tin cơ bản cần thiết về các hệ thống phát hiện xâm nhập dựa trên sự bất thường. Sau đó, chúng tôi khám phá công việc liên quan trong miền ngưỡng thích ứng. Bây giờ chúng tôi cung cấp thông tin cơ bản về ADS.

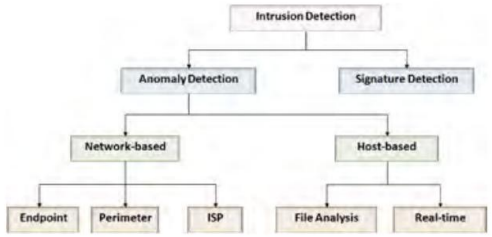
2.1. Hệ thống

phát hiện xâm nhập nền (IDS) thường được phân loại thành hai loại chính - phát hiện lạm dụng (còn được gọi là phát hiện chữ ký) và phát hiện bất thường. Việc phát hiện sử dụng sai yêu cầu phải có chữ ký trước và nó được sử dụng rộng rãi trong các chương trình chống vi-rút ngày nay. Mặc dù các phương pháp dựa trên chữ ký cung cấp tỷ lệ phát hiện 100% mà không có cảnh báo sai đối với các mối đe dọa đã biết (các mối đe dọa đã trích xuất chữ ký), các phương pháp này không phát hiện được các cuộc tấn công zero-day. Tốc độ gia tăng nhanh chóng của các cuộc tấn công zero-day đã tạo thêm một lớp phức tạp cho vấn đề này và gây khó khăn cho việc trích xuất và phổ biến chữ ký mối đe dọa một cách kịp thời. Mặt khác, một máy dò dị thường lập mô hình hành vi bình thường/lành tính và kiểm tra độ lệch của một phiên bản chưa biết so với mô hình đó. Nếu độ lệch so với hành vi bình thường vượt quá một ngưỡng cụ thể, phiên bản không xác định được đánh dấu là bất thường hoặc độc hại. Hoạt động của một hệ thống phát hiện bất thường được thể hiện trong Hình 1. Hình 1 cho thấy sự gia tăng lưu lượng truy cập trên cổng 80 do sâu CodeRed [Moore et al. 2002]. Nếu số lượng máy chủ duy nhất có lưu lượng truy cập trên cổng 80 được vẽ trên biểu đồ và giá trị ngưỡng thích hợp được chọn (hơn 800 máy chủ trong trường hợp này), người ta có thể thấy rõ sự bất thường.

Phát hiện bất thường có thể được phân loại thành các công cụ phát hiện bất thường dựa trên mạng và máy chủ tùy thuộc vào việc các công cụ phát hiện có hoạt động trên lưu lượng mạng hay không



Hình 1. Hoạt động của hệ thống phát hiện bất thường - CodeRed tăng đáng kể lưu lượng truy cập trên cổng 80 có thể được đặt ngưỡng để phát hiện bất thường [Moore et al. 2002].



Hình 2. Phân loại cấp cao của các hệ thống phát hiện xâm nhập hiện có.

hoặc đặc điểm vật chủ. Các trình phát hiện bất thường dựa trên mạng hoạt động trên nhiều tính năng lưu lượng khác nhau và các trình phát hiện bất thường dựa trên máy chủ tương tự phân tích tĩnh/động một tệp thực thi hoặc phân tích hành vi của máy chủ để phát hiện sự bất thường. Một ví dụ về phân loại IDS được hiển thị trong Hình 2.

Độ chính xác của các hệ thống phát hiện xâm nhập thường được đánh giá trên hai yếu tố cạnh tranh nhau, đó là tỷ lệ phát hiện (True Positive, TP) và tỷ lệ cảnh báo sai (False Positive, FP). Tỷ lệ phát hiện được tính bằng cách quan sát các báo động thực sự được tạo bởi máy dò, nghĩa là máy dò đã phân loại một sự kiện nguy hiểm là độc hại và tỷ lệ cảnh báo sai được tính bằng cách quan sát các thông báo dương tính giả do máy dò tạo ra, tức là máy dò phân loại một sự kiện lành tính là một sự kiện độc hại.

Nói chung, các hệ thống phát hiện bất thường có một tham số có thể điều chỉnh (hoặc tập hợp các tham số) để điều chỉnh tốc độ phát hiện và tốc độ cảnh báo sai được gọi là ngưỡng. Điều chỉnh ngưỡng thành các giá trị khác nhau dẫn đến một số tỷ lệ phát hiện so với tỷ lệ cảnh báo sai. Giá trị ngưỡng được thay đổi để có được các tỷ lệ phát hiện khác nhau so với các tỷ lệ cảnh báo sai khác nhau. Sau đó, tỷ lệ phát hiện được vẽ trên trục y và tỷ lệ cảnh báo sai được vẽ trên trục x và đường nối các điểm này tạo ra một đường cong được gọi là đường cong Đặc tính hoạt động của máy thu (ROC). Đường cong ROC được sử dụng như một tiêu chuẩn thực tế để trực quan hóa việc đánh giá độ chính xác của máy dò dị thường. Đường cong càng dốc (diện tích dưới đường cong càng nhiều) thì độ chính xác của máy dò càng cao. Công việc của chúng tôi chủ yếu tập trung vào các thiết bị phát hiện bất thường đang hoạt động theo phương pháp ngưỡng.

2.2. Công việc có liên quan

Một số công cụ phát hiện bất thường mạng gần đây cung cấp các phương pháp tính ngưỡng cho các thuật toán cụ thể của chúng [Ide và Kashima 2004; Jung et al. 2004; Lakhina và cộng sự. 2004]. Ví dụ, Ide và Kashima [2004], sử dụng tính năng cụ thể của ADS, chẳng hạn như thước đo liên quan đến số lượng điểm dữ liệu trong một ranh giới tới hạn nhất định và thước đo kích thước hiệu quả của từng cụm riêng để tính toán ngưỡng. Tương tự, Jung et al. [2004] lập mô hình bước đi ngẫu nhiên cho xác suất du ngoạn và xác suất du ngoạn này được lấy từ hai bộ xác suất có thể có dành riêng cho nguyên tắc phát hiện của chúng. Lakhina và cộng sự. [2004] cũng sử dụng các thành phần chính để tính toán ngưỡng, đó là các tính năng của máy dò được đề xuất. Vì hầu hết các nghiên cứu này không phục vụ cho hành vi thay đổi theo thời gian của đầu vào, do đó các phương pháp này không mang lại hiệu suất chấp nhận được trong các điều kiện giao thông khác nhau [Ashfaq et al. 2008].

Một số sản phẩm ADS thương mại cũng điều chỉnh ngưỡng của chúng phù hợp với đặc điểm lưu lượng đầu vào [Arbor PeakFlow; Bảo vệ dị thường của Cisco]. Vì các sản phẩm này sử dụng các thuật toán độc quyền nên chúng tôi không thể chứng minh mức độ chính xác đạt được

bởi các thuật toán này. Vì vậy, trong phần này, chúng tôi chỉ thảo luận về các cách tiếp cận do cộng đồng nghiên cứu đề xuất.

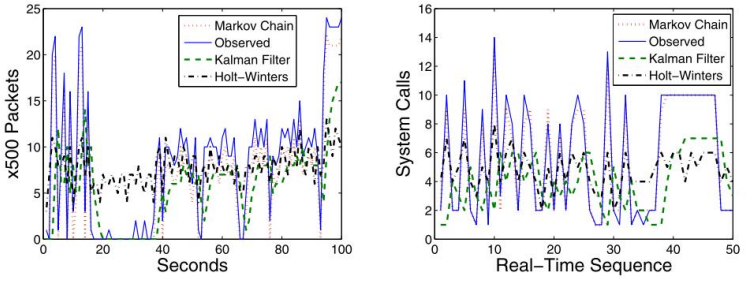
Một thuật toán điều chỉnh mô hình đã được đề xuất trong Yu et al. [2007], tự động điều chỉnh mô hình phát hiện của IDS trong thời gian thực và đạt được tối đa 20% mỗi cải tiến về hình thức. Tuy nhiên, mô hình này phụ thuộc vào phản hồi của người vận hành hệ thống, do đó cần có sự can thiệp của con người. Một phần mở rộng của Yu et al. [2007] đã được đề xuất trong Yu et al. [2008], sử dụng bộ lọc dự đoán để chọn một cách thích ứng các dự đoán đáng ngờ nhất để người vận hành hệ thống có thể xác minh các dự đoán này.
Các đánh giá cho thấy trong số 304 dự đoán, 100 dự đoán là báo động sai. Cách tiếp cận này giảm thiểu tải cho người vận hành hệ thống, tuy nhiên nó vẫn cần sự can thiệp của con người.

Một mô hình đánh đổi hiệu suất cho ADS đã được đề xuất trong Cardenas et al. [2006], khám phá đường cong Đặc điểm hoạt động phát hiện xâm nhập (IDOC) như một khuôn khổ hợp lệ để đánh giá sự cân bằng hiệu suất cho ADS. IDOC mô hình hóa vấn đề đánh giá dưới dạng vấn đề tối ưu hóa đa tiêu chí bao gồm các chỉ số về chi phí và độ nhạy dự kiến. Đường cong IDOC kết hợp tất cả các tham số có liên quan ảnh hưởng đến hiệu suất thực tế của IDS. Tuy nhiên, một mô hình đánh giá khác đã được đề xuất [Ryu và Rhee 2008] cho các thiết bị phát hiện bất thường dưới sự ràng buộc của việc kiểm tra, đó là giới hạn của nhân viên an ninh thông tin trong việc kiểm tra sự cố. Công việc thảo luận về phân tích để giúp quản trị viên hệ thống hiểu được các khía cạnh vận hành, quản lý và đánh giá của một IDS. Mặc dù cả hai cách tiếp cận này đều cung cấp các biện pháp thay thế để đánh giá các ADS khác với ROC, nhưng các cách tiếp cận này không cung cấp cơ chế để tự động hóa hoàn toàn quy trình ADS.

Công việc trước đây thích hợp nhất trong lĩnh vực này là Agosta et al. [2007] và Gu et al. [2006]. Một ADS đã được đề xuất trong Agosta et al. [2007] điều chỉnh ngưỡng cũ của nó theo các biến thể quan sát được trong đầu vào. Tuy nhiên, một kỹ thuật thích ứng ngưỡng thực tế thay vì được phát minh cho một thuật toán phát hiện bất thường cụ thể sẽ hoạt động trơn tru và cung cấp các cải tiến về độ chính xác cho thuật toán hiện có hoạt động theo cách ngưỡng. Gu et al. [2006] đã đề xuất một tập hợp các phép đo lý thuyết thông tin có thể đo lường định lượng hiệu quả của IDS về khả năng biểu diễn tính năng, mất thông tin phân loại và khả năng phát hiện xâm nhập tổng thể. Khung đề xuất cung cấp các hướng dẫn thiết thực để tinh chỉnh IDS (tĩnh và thời gian thực), đánh giá IDS và thiết kế IDS.

Tuy nhiên, nó sử dụng phương pháp tính toán khả năng phát hiện xâm nhập và các tính năng khác của máy dò để xác định ngưỡng tại thời điểm nhất định. Mặt khác, chúng tôi mô hình hóa các điểm bất thường được máy dò dị thường quan sát được trong thời gian thực.

Các kỹ thuật tương ứng với công việc của chúng tôi là các kỹ thuật luồng dữ liệu trôi dạt khái niệm [Aggarwal et al. 2006; Cretu-Ciocarlie et al. 2009; Kolter và Maloof 2005; Masud et al. 2010, 2011]. Một thuật toán phát hiện lớp mới cùng với phân loại được đề xuất trong Masud et al. [2011]. Nó có thể sử dụng bất kỳ bộ học cơ sở nào với ít hoặc không cần sửa đổi để tạo và lưu ranh giới quyết định được sử dụng để phân loại sau này. Một phương pháp phát hiện lớp mới lạ khác đã được đề xuất trong Masud et al. [2010]. Nó sử dụng ngưỡng thích ứng và Hệ số Gini để phát hiện ngoại lệ như một phần của phát hiện lớp mới. Một khung làm việc để đánh giá sự trôi dạt khái niệm và các phương pháp tập hợp để xử lý sự trôi dạt khái niệm đã được đề xuất trong Bifet et al. [2009]. Trong Cretu-Ciocarlie et al. [2009], một khung tự động dựa trên khả năng tự hiệu chuẩn được trình bày, không thể tin được đối với các cảm biến phát hiện bất thường. Nó sử dụng phương pháp tập hợp nhưng với học tập không giám sát. Hơn nữa, nó xây dựng các mô hình vi mô cho mục đích thích ứng, sau đó được sử dụng để phát hiện bất thường bằng phương pháp bỏ phiếu. Chen và cộng sự. [2008] khai thác các khái niệm ngoại tuyến từ một luồng và xây dựng các mô hình chất lượng cao cho từng mô hình. Nó tìm thấy mô hình hiện tại có khả năng nhất và phân loại bằng cách sử dụng nó. Khung dựa trên tính trung bình và biểu quyết đơn giản/mô hình được trình bày trong Gao et al. [2007]. Tất cả các cách tiếp cận này xử lý thay đổi hành vi thời gian chạy trong dữ liệu và cập nh



(a) LBNL background traffic (b) UNM System Calls Sequence

Hình 3. Độ chính xác dự đoán của Markov Chain đối với các đặc điểm dữ liệu đầu vào khác nhau của ADS; vì tốc độ gói trong (a) được chia thành các thùng có kích thước bằng nhau là 500, 0 (trên trục y) biểu thị tốc độ gói trong khoảng từ 0 đến 499 gói/giây.

phân loại cơ bản/mô hình cơ sở. Tuy nhiên, cách tiếp cận của chúng tôi điều chỉnh ranh giới của mô hình phân loại. Hầu hết các cách tiếp cận dựa trên tập hợp dựa trên hiện tượng đa lớp, tuy nhiên, không phải tất cả các bộ phát hiện xâm nhập đều xây dựng mô hình đa lớp. Do đó, các phương pháp tiếp cận theo khái niệm có thể yêu cầu sửa đổi các bộ phát hiện xâm nhập để được sử dụng làm người học cơ sở. Tuy nhiên, ngưỡng thích ứng không yêu cầu sửa đổi trong các thiết bị phát hiện xâm nhập và coi chúng như hộp đen. Sau đó, chúng tôi sẽ chỉ ra rằng các yêu cầu tính toán đối với ngưỡng thích ứng ít hơn nhiều so với các kỹ thuật trôi dạt khái niệm. Phương pháp tiếp cận khái niệm trôi dạt và kỹ thuật ngưỡng quảng cáo có thể được triển khai song song với (các) sửa đổi nhỏ.

3. ĐỘNG LỰC VÀ PHƯƠNG PHÁP

Trong phần này, chúng tôi thảo luận và nhấn mạnh cơ sở lý luận và phương pháp luận cấp cao trong công việc của chúng tôi.

3.1. Động lực

Độ chính xác của ADS được đặc trưng theo truyền thống trên mặt phẳng ROC bằng cách áp dụng các ngưỡng phân loại khác nhau cho điểm số bất thường của ADS. Các điểm trên mặt phẳng ROC thu được bằng cách tính tỷ lệ phát hiện trung bình (trục y) so với tỷ lệ cảnh báo sai trung bình (trục x) cho từng giá trị ngưỡng. Độ dốc và chiều cao của đường cong ROC định lượng độ chính xác mà ADS có thể đạt được. Mặc dù đánh giá ROC rất hữu ích để hiểu được độ chính xác trong trường hợp trung bình của ADS, nhưng các phương pháp để đạt được điểm chính xác ROC tốt nhất cho dữ liệu tùy ý (lưu lượng truy cập hoặc hệ điều hành) không được nghiên cứu kỹ.

Trong thực tế, một ADS sẽ phải bằng cách nào đó tìm hiểu ngưỡng phân loại tốt cho một hành vi lành tính tùy ý trong thời gian thực. Tệ hơn nữa, dữ liệu thô được nhập vào ADS thường hiển thị các biến thể đáng kể. Các đặc điểm lưu lượng truy cập khác nhau giữa các tổ chức và điểm triển khai mạng và do các mô hình sử dụng hàng ngày và khác. Ví dụ, hãy xem xét tốc độ lưu lượng truy cập nền LBNL được hiển thị trong Hình 3(a) (đường liền nét). Có thể quan sát thấy rằng tốc độ lưu lượng thay đổi từ khoảng 500 pkts/giây thành 10.000 pkts/giây trong vòng vài giây. Tương tự, các chỉ số phát hiện bất thường dựa trên máy chủ là một chức năng của hành vi người dùng, ứng dụng đang được sử dụng, hệ điều hành, phần cứng, v.v. Ví dụ: hãy xem xét các cuộc gọi hệ thống bình thường được thực hiện trong Hình 3(b). Có thể thấy rằng các cuộc gọi hệ thống thay đổi thất thường từ 1 đến 15 trong các dấu vết lành tính. Có thể lập luận bằng trực giác rằng một giá trị cố định (thời gian và

(a) Kullback-Leibler Detector on UNM Dataset

(b) Maximum Entropy Detector on LBNL Dataset

Hình 4. Í t dao động hơn trong điểm bất thường đối với các biến thể trong đặc tính đầu vào (lưu lượng truy cập hoặc máy chủ); do đó, điểm bất thường tương đối dễ dự đoán hơn.

không thể đạt được độ chính xác tốt cho đầu vào thay đổi theo thời gian như vậy.

Cuối cùng, khi các đặc điểm dữ liệu đầu vào khác nhau, việc xác định ngưỡng cố định yêu cầu can thiệp thủ công lặp đi lặp lại. Trong một kịch bản hoạt động điển hình, quản trị viên hệ thống/mạng chịu trách nhiệm điều chỉnh độ nhạy của bộ phát hiện bất thường dựa trên mạng khi số lượng cảnh báo sai (nghĩa là lưu lượng được phân loại là độc hại nhưng thực tế là lành tính) tăng lên. Tương tự, các ADS dựa trên máy chủ yêu cầu người dùng điều chỉnh độ nhạy của nó để phục vụ cho các yêu cầu hành vi và bảo mật của họ. Việc nhập thủ công lặp đi lặp lại như vậy làm cho ADS ít tự động hơn và dễ bị lỗi cấu hình hơn. Hơn nữa, trong một hệ thống thời gian thực, rất khó để xác định xem ngưỡng được cấu hình thủ công có mang lại độ chính xác tốt hay không.

3.2. Phương pháp luận

chung Chúng tôi lập luận rằng một ADS hiệu quả sẽ tự động phát hiện các mẫu dữ liệu đầu vào khác nhau và điều chỉnh ngưỡng phân loại của nó cho phù hợp. Nếu chính xác, cơ chế ngưỡng thích ứng như vậy sẽ cho phép ADS đạt được điểm hoạt động tốt trên mặt phẳng ROC. Là một sản phẩm phụ, ngưỡng thích ứng cũng sẽ giảm nhu cầu điều chỉnh ngưỡng của con người, do đó làm cho ADS tự động hơn.

Chúng tôi quan sát thấy rằng các biến thể đáng kể trong đặc điểm dữ liệu đầu vào rất khó theo dõi. Thay vào đó, việc theo dõi điểm bất thường của ADS trước khi áp dụng chức năng ngưỡng sẽ dễ dàng hơn nhiều. Ví dụ, Hình 4 cho thấy các biến thể ở đầu vào và đầu ra của máy chủ và ADS mạng. Có thể quan sát thấy từ Hình 4(a) rằng, trong khi đầu vào (các cuộc gọi hệ thống) thay đổi thường xuyên từ 1 đến 160, thì không có nhiều thay đổi được quan sát thấy ở đầu ra (điểm bất thường). Tương tự như vậy, Hình 4(b) cho thấy những thay đổi đáng kể được quan sát thấy trong các đặc điểm lưu lượng truy cập, nhưng những thay đổi thất thường như vậy không được phản ánh trong điểm số bất thường. Do đó, điểm bất thường dễ theo dõi hơn vì chúng giảm dữ liệu đầu vào nhiều chiều thành một tập hợp điểm tương đối nhỏ. Ngoài lợi thế về độ phức tạp, do các điểm số này nhất quán với các đặc điểm của dữ liệu đầu vào và vì các điểm số này bao gồm miền của hàm ngưỡng, nên theo trực giác có khả năng là các điểm số bất thường theo dõi thích ứng mang lại độ chính xác cao hơn so với theo dõi trực tiếp dữ liệu đầu vào. Luận điểm này dẫn chúng ta đến cơ sở lý luận sau cho kỹ thuật ngưỡng thích ứng được đề xuất trong bài viết này: Nếu chúng ta có thể dự đoán chính xác các giá trị dự kiến của điểm số bất thường trong tương lai trong các điều kiện thuận lợi, thì ngưỡng phân loại có thể được điều chỉnh như một hàm của điểm số dự đoán.

Theo dõi điểm số bất thường của ADS yêu cầu một mô hình điểm số mạnh mẽ được quan sát trong điều kiện bình thường. Để phát triển một mô hình như vậy, trong các phần sau, chúng tôi đánh giá một số thuộc tính thống kê thích hợp của điểm bất thường.

4. BỘ DỮ LIỆU VÀ HỆ THỐNG PHÁT HIỆN XÂM NHẬP DỰA TRÊN SỰ THẤT THƯỜNG

Bây giờ chúng ta thảo luận ngắn gọn về bộ dữ liệu mạng và máy chủ cũng như ADS được sử dụng để đánh giá. Chi tiết về các bộ dữ liệu và ADS này được cung cấp tương ứng trong Phụ lục A và Phụ lục B.

4.1. Bộ dữ liệu

Để cho thấy hiệu quả của phương pháp được đề xuất đối với cả ADS dựa trên mạng và máy chủ, chúng tôi sử dụng bốn bộ dữ liệu dựa trên mạng (chứa lưu lượng truy cập mạng) và ba bộ dữ liệu dựa trên máy chủ (chứa các lệnh gọi hệ thống/chuỗi API). Tất cả các bộ dữ liệu đã được sử dụng trong nghiên cứu này đã được thu thập độc lập, dán nhãn và có sẵn công khai, ngoại trừ một bộ dữ liệu. Hơn nữa, lựa chọn bộ dữ liệu của chúng tôi đảm bảo tính đa dạng. Đối với dữ liệu lưu lượng truy cập mạng, chúng tôi đã sử dụng các bộ dữ liệu sử dụng các điểm thu thập khác nhau trong một công việc mạng. Một mặt, chúng tôi sử dụng bộ dữ liệu từ cổng và bộ định tuyến trung gian, mặt khác, chúng tôi sử dụng lưu lượng cấp phiên được thu thập tại nhiều máy chủ cuối. Đối với bộ dữ liệu dựa trên bộ định tuyến cổng, chúng tôi sử dụng bộ dữ liệu LBNL [Bộ dữ liệu LBNL] và bộ dữ liệu ISP. Bộ dữ liệu LBNL được thu thập tại bộ định tuyến cổng của Phòng thí nghiệm quốc gia Lawrence Berkeley (LBNL), Hoa Kỳ. Lưu lượng tấn công bao gồm các lần quét cổng đến được nhắm mục tiêu tới các máy chủ nội bộ có tốc độ thấp so với lưu lượng nền (lành tính).

Bộ dữ liệu ISP được thu thập tại cổng của Nhà cung cấp dịch vụ Internet (ISP) hàng đầu ở Pakistan vào năm 2012 cho công việc này; do đó, nó đại diện cho mạng lưới người dùng khác nhau, thực tế và gần đây. Các gói hoàn chỉnh bao gồm tải trọng đã bị bắt. Lưu lượng tấn công bao gồm mật khẩu brute force, SQL injection, leo thang đặc quyền, tải lên tệp script, khởi động shell ngược. Chúng tôi cũng đã sử dụng bộ dữ liệu dựa trên bộ định tuyến trung gian để xử lý lưu lượng truy cập từ nhiều phòng thí nghiệm nghiên cứu [Bộ dữ liệu NUST] đại diện cho nhiều người dùng. Lưu lượng tấn công bao gồm quét cổng và tấn công tràn ngập. Bộ dữ liệu gần đây cũng được thu thập trên một mạng lớn vào năm 2009 và lần đầu tiên được sử dụng trong Ali et al.

[2010]. Cuối cùng, chúng tôi sử dụng bộ dữ liệu Điểm cuối, chứa lưu lượng truy cập cấp phiên được thu thập tại nhiều máy chủ đã bị lây nhiễm bằng các loại phần mềm độc hại quét cổng khác nhau [WisNet ADS].

Để đánh giá các ADS dựa trên máy chủ, chúng tôi sử dụng các lệnh gọi hệ thống Linux (dấu vết tổng hợp và theo kinh nghiệm) và các lệnh gọi trình tự API của Windows. Dấu vết tổng hợp từ UNM [Bộ dữ liệu UNM] được tạo ra bằng cách liệt kê các nguồn biến thể tiềm năng cho các hoạt động gửi thư bình thường. Chúng tôi đã sử dụng dấu vết xâm nhập của các vòng sscp, giải mã và chuyển tiếp được cung cấp tại [Bộ dữ liệu UNM]. Một tập dữ liệu khác mà chúng tôi sử dụng được thu thập tại Phòng thí nghiệm MIT Lincoln [MIT Dataset]. Bộ dữ liệu MIT LL bao gồm các cuộc gọi hệ thống của máy chủ Solaris.
Từ bộ dữ liệu MIT LL, chúng tôi đã sử dụng dữ liệu từ tuần đầu tiên (bắt đầu từ ngày 01/03/1999) dưới dạng tập hợp hoàn hảo và đối với các cuộc tấn công được dán nhãn, dữ liệu của tuần thứ hai (bắt đầu từ ngày 08/03/1999) đã được sử dụng. Cuối cùng, chúng tôi đã sử dụng bộ dữ liệu nhật ký cuộc gọi API được tạo bằng 416 phần mềm độc hại (trojan, vi rút và sâu) và 100 tệp thực thi Win32 lành tính [Bộ dữ liệu Nexgin]. Trình theo dõi cuộc gọi API thương mại đã được sử dụng để ghi nhật ký các lệnh gọi API được thực hiện bởi mỗi quy trình bằng cách chèn móc nối chế độ nhân cho các quy trình đang chạy trên Microsoft Windows XP.

4.2. Bộ phát hiện bất

thường Trước khi mô tả các ADS thời gian thực được sử dụng trong công việc này, chúng tôi xin nhắc lại rằng thuật toán thích ứng ngưỡng thực tế không nên dành riêng cho một ADS cụ thể.
Do đó, mặc dù chúng tôi đã chọn một số ADS để đo điểm chuẩn hiệu suất và bằng chứng về khái niệm, nhưng tất cả phân tích và mô tả đặc điểm sẽ được cung cấp trong

các phần sau đây là chung chung và phải phù hợp với các ADS hoạt động theo nguyên tắc ngưỡng.

Chúng tôi cũng nhấn mạnh rằng các ADS được sử dụng trong nghiên cứu này khá đa dạng về các tính năng và nguyên tắc phát hiện nói dối. Chẳng hạn, các ADS mạng được sử dụng trong công việc này bao gồm các hệ thống mô hình hóa quy tắc đơn giản như PHAD [Mahoney và Chan 2001] cho đến các hệ thống tự học phức tạp như Maximum-Entropy [Gu et al. 2005], Thử nghiệm giả thuyết tuần tự [Jung et al. 2004] và PAYL [Wang và Stolfo 2004] phát hiện bất thường giao thông. Tương tự, về phía máy chủ, chúng tôi bao gồm một trình phát hiện trình tự dị thường đơn giản [Forrest et al. 1996], một máy dò dựa trên máy học [Kang et al. 2005] và các máy dò dựa trên căn chỉnh trình tự [Gao et al. 2005; Sung et al. 2004]. Sự đa dạng này được giới thiệu để cho thấy rằng bất kỳ ADS thời gian thực nào, bất kể chức năng của nó là gì, đều có thể thích ứng với các biến thể trong đầu vào của nó để cung cấp hiệu suất chấp nhận được mà không cần cấu hình thủ công đáng kể.

Tất cả các ADS được sử dụng trong công việc này, đã được đào tạo bằng cách sử dụng 50% tổng số dữ liệu; 50% còn lại được sử dụng để thử nghiệm. Các mẫu giữa tập huấn luyện và tập kiểm tra không trùng nhau. Để đo điểm chuẩn hiệu suất, chúng tôi thay đổi ngưỡng phát hiện bất thường cũ của tất cả các máy dò để tạo đường cong ROC. Tất cả các tham số khác của ADS giống như được báo cáo trong các bài báo gốc. Bây giờ chúng tôi thảo luận ngắn gọn về các ADS được sử dụng trong công việc này; độc giả được tham khảo các bài báo gốc để biết thêm chi tiết về các thuật toán này.

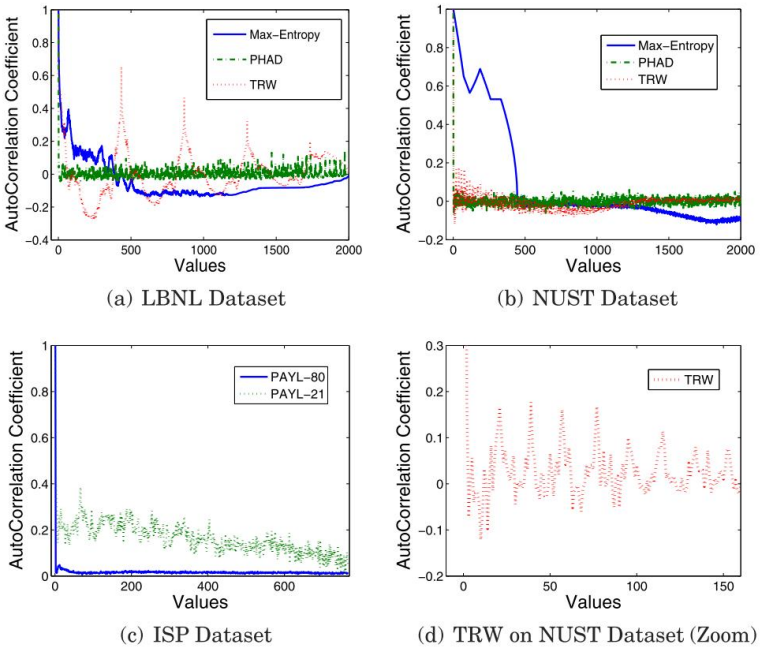
Máy dò Entropy cực đại [Gu et al. 2005] ước tính phân bổ lưu lượng thuận lợi bằng cách sử dụng ước tính entropy cực đại và so sánh nó với phân bổ lưu lượng theo cửa sổ thời gian thực sử dụng thước đo phân kỳ Kullback-Leibler (KL). Một cảnh báo được đưa ra nếu phân kỳ KL của lớp gói vượt quá ngưỡng. Thuật toán TRW [Jung et al. 2004] phát hiện các lần quét cổng đến bằng cách kiểm tra giả thuyết rằng xác suất nỗ lực kết nối thành công sẽ cao hơn nhiều đối với máy chủ lành tính so với máy quét. Packet Header Anomaly Detector (PHAD) [Mahoney và Chan 2001] tìm hiểu phạm vi giá trị bình thường cho 33 trường trong tiêu đề gói. Điểm bất thường được chỉ định cho từng trường tiêu đề gói trong giai đoạn thử nghiệm và điểm số trường được tính tổng để có được điểm bất thường tổng hợp của gói. Cuối cùng, Payload-based de tector (PAYL) [Wang và Stolfo 2004] mô hình hóa tải trọng ứng dụng bình thường của lưu lượng công việc mạng. Nó xây dựng một cấu hình bình thường bằng cách tính toán phân phối tần số byte và độ lệch chuẩn của tải trọng ứng dụng chảy đến một máy chủ và cổng duy nhất.

Cấu hình bình thường này được so sánh với cấu hình thời gian chạy sử dụng khoảng cách Mahalanobis. Sẽ có cảnh báo nếu khoảng cách tăng hơn giá trị ngưỡng.

Nhúng độ trễ thời gian trình tự (STIDE) [Forrest et al. 1996] xây dựng một hồ sơ lành tính cho một quá trình bằng cách sử dụng tất cả các chuỗi gọi hệ thống liền kề duy nhất có độ dài cố định và được xác định trước. Trình phát hiện Máy Vector Hỗ trợ (SVM) [Kang et al. 2005] xem xét một loạt các cuộc gọi hệ thống mà không tính đến trình tự hoặc thứ tự của các cuộc gọi hệ thống. LƯU [Sung et al. 2004] tìm thấy mối tương quan giữa các biến thể của phần mềm độc hại bằng cách căn chỉnh chuỗi lệnh gọi API và sau đó sử dụng ba chức năng tương tự để xác định xem tệp không xác định có phải là biến thể của phần mềm độc hại đã biết hay không. Bdist [Gao et al. 2005] phát hiện các biến thể của một phần mềm độc hại đã biết bằng cách tìm sự giống nhau/khác biệt về ngữ nghĩa giữa hai bản sao của phần mềm độc hại bằng cách sử dụng khoảng cách tiến hóa khi đầu vào của các quy trình giống nhau.

5. ĐIỀU TRA SƠ BỘ CÁC ĐIỂM BẤT THƯỜNG

Trong phần này, chúng tôi đánh giá các thuộc tính thống kê của điểm số bất thường của ADS có thể được sử dụng để tự động lập mô hình và điều chỉnh ngưỡng phân loại của nó. Dựa trên các thuộc tính thống kê này, phần tiếp theo đề xuất một thuật toán ngưỡng thích ứng có thể theo dõi chính xác hành vi thay đổi của lưu lượng truy cập thời gian thực và/hoặc hành vi của hệ điều hành.



(a) LBNL Dataset (b) NUST Dataset

(c) ISP Dataset (d) TRW on NUST Dataset (Zoom)

Hình 5. Xu hướng suy giảm của Hệ số tự tương quan của các điểm bất thường của ADS Mạng thể hiện mức độ phụ thuộc thời gian nhất định; Điểm của ADS có Entropy cực đại được vẽ cho cổng 80.

5.1. Sự phụ thuộc tạm thời vào điểm số bất thường Chúng tôi đã

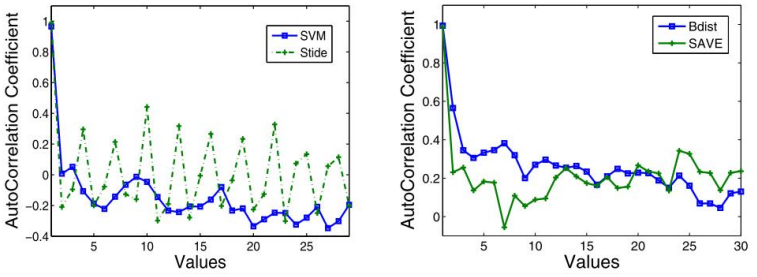
phân tích một số thuộc tính thống kê của lưu lượng truy cập bình thường và điểm số hệ điều hành. Một thuộc tính có liên quan cung cấp cho chúng tôi những hiểu biết thú vị về điểm số bất thường là phân tích sự phụ thuộc theo thời gian của chúng. Có thể lập luận bằng trực giác rằng, miễn là dữ liệu đầu vào của ADS được tạo ra bởi một nguồn lành tính, điểm số bất thường quan sát được ở đầu ra của ADS sẽ thể hiện một mức độ phụ thuộc nhất định về thời gian. Trong trường hợp có sự bất thường, nhiễu loạn trong cấu trúc phụ thuộc này được đánh dấu là bất thường. Do đó, mức độ phụ thuộc vào thời gian có thể đóng vai trò là thước đo quan trọng để mô hình hóa các điểm bất thường.

Tự tương quan đo lường sự phụ thuộc thời gian trên mức trung bình giữa các biến dom đã chạy trong một quy trình ngẫu nhiên tại các thời điểm khác nhau. Đối với độ trễ k cho trước, hàm tự tương quan của quá trình ngẫu nhiên Xn (trong đó n là chỉ số thời gian) được định nghĩa là:

$$\rho[k] = \frac{E\{X0Xk\} \quad E\{X0\}E\{Xk\}}{\sigma X0 \ \sigma Xk}, \tag{1}$$

trong đó E{.} đại diện cho phép toán kỳ vọng và σXk là độ lệch chuẩn của biến ngẫu nhiên ở độ trễ thời gian k. Giá trị của hàm tự tương quan nằm trong khoảng [1, 1], trong đó ρ[k] = 1 nghĩa là tương quan hoàn hảo ở độ trễ k (điều này hiển nhiên đúng với k = 0) và ρ[k] = 0 nghĩa là không có tương quan ở độ trễ k.

Hình 5 và 6 hiển thị chức năng tự tương quan được vẽ tương ứng với độ trễ đối với điểm ADS dựa trên mạng và máy chủ. Đối với tất cả các ADS, có thể dễ dàng quan sát thấy mức độ phụ thuộc thời gian cao ở độ trễ nhỏ. Mối tương quan này giảm dần theo thời gian và cuối cùng giảm xuống một giá trị không đáng kể. Tuy nhiên, sự phân rã tương quan không nhất quán đối với tất cả các ADS.



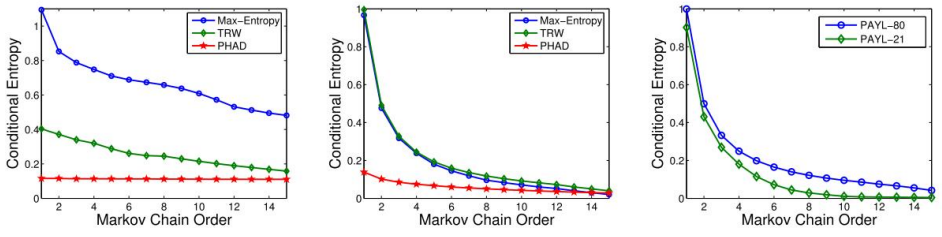
(a) UNM Dataset (b) Nexgin Dataset

Hình 6. Xu hướng suy giảm của Hệ số tự tương quan của các điểm bất thường của ADS máy chủ thể hiện mức độ phụ thuộc thời gian nhất định.

Hình 5(a), (b) và (c) hiển thị sự phân rã tương quan đối với các ADS dựa trên mạng trên bộ dữ liệu LBNL, NUST và ISP, tương ứng. Tuy nhiên, Hình 5(d) hiển thị phiên bản phóng to của TRW trên NUST từ Hình 5(b). Có thể thấy rằng PHAD cho thấy mức độ phân rã tương quan cao nhất trên bộ dữ liệu LBNL và NUST; nghĩa là, một điểm bất thường của PHAD chỉ tương quan với một vài điểm trước đó. Điều này có thể được giải thích bằng cách lưu ý rằng thuật toán phát hiện bất thường của PHAD không phụ thuộc vào tốc độ/khối lượng hoặc tần suất lưu lượng. Thay vào đó, PHAD tự phân loại từng gói mà không mượn bất kỳ thông tin nào từ các gói trước đó. Trình phát hiện Entropy cực đại cho thấy sự phân rã tương quan chậm nhất trên cả bộ dữ liệu LBNL và NUST vì thuật toán phát hiện bất thường của nó hoạt động trên phân phối sử dụng cổng. Tần số cổng không thay đổi đáng kể theo thời gian (như đã được trình bày trong Lakhina và cộng sự [2005]), và do đó, mỗi điểm bất thường của máy dò Entropy cực đại vốn đã tương quan với các điểm trước đó. Xu hướng tương tự cũng được quan sát đối với TRW ADS. Thật thú vị, khi TRW được đánh giá ở độ trễ lớn trên bộ dữ liệu LBNL, nó cho thấy cấu trúc tương quan có tính chu kỳ trong đó mối tương quan thể hiện xu hướng lặp lại từ cao xuống thấp (thay vì thể hiện xu hướng giảm dần). Tuy nhiên, trên tập dữ liệu NUST, nó giảm xuống ngay lập tức giống như PHAD nhưng vẫn có thể quan sát thấy xu hướng định kỳ ở quy mô nhỏ hơn khi nhìn kỹ hơn như trong Hình 5(d), là phiên bản phóng to của Hình 5(b). Tương tự, có thể quan sát thấy xu hướng giảm dần đối với PAYL trên bộ dữ liệu ISP. Vì PAYL cũng tự phân loại từng gói giống như PHAD nên mức độ phân rã rất cao. Xu hướng giảm dần của PAYL trên bộ dữ liệu ISP được hiển thị cho cổng 80 và 21, được ký hiệu lần lượt là PAYL-80 và PAYL-21.

Hình 6(a) và (b) lần lượt hiển thị xu hướng tương quan cho các ADS dựa trên máy chủ trên bộ dữ liệu UNM và Nexgin. Có thể thấy rõ trong Hình 6(a) rằng cả STIDE và SVM đều thể hiện sự phân rã theo cấp số nhân trên tập dữ liệu UNM vì cả hai ADS đều hoạt động trên các khối lệnh gọi hệ thống mà không mượn nhiều thông tin từ các chuỗi lệnh gọi trước đó.
Một xu hướng phân rã theo cấp số nhân rất giống nhau được hiển thị cho Bdist và SAVE trên tập dữ liệu Nexgin trong Hình 6(b). Xu hướng tương quan giảm dần là một thuộc tính phổ biến phát sinh do hậu quả của bản chất Markovian của điểm số bất thường. Nói cách khác, bất kể ADS là gì, những thay đổi về xu hướng lưu lượng truy cập nền/tính năng của hệ điều hành gây ra sự tương quan đáng kể về điểm số bất thường theo thời gian. Ví dụ: tại một thời điểm nhất định, lưu lượng truy cập thuộc các phiên tương tự hoặc lệnh gọi hệ thống thuộc các ứng dụng tương tự được quan sát. Tuy nhiên, theo thời gian, các ứng dụng/kết nối mạng mới được khởi chạy, do đó thay đổi hành vi của đầu vào thành ADS gây ra sự giảm tương quan trong điểm số bất thường.

Dựa trên cấu trúc tương quan giảm dần hiện diện trong các điểm số bất thường, chúng tôi lập luận rằng có thể sử dụng mô hình ngẫu nhiên của một số điểm số trước đó để dự đoán chính xác các điểm số trong tương lai. Tuy nhiên, để hạn chế sự phức tạp của việc thích ứng ngưỡng, chúng ta phải trả lời câu hỏi sau: Cần bao nhiêu điểm bất thường trong quá khứ để dự đoán chính xác điểm tiếp theo? Thật thú vị, phần sau đây cho thấy rằng câu trả lời cho



(a) LBNL Dataset (b) NUST Dataset (c) ISP Dataset

Hình 7. Xu hướng suy giảm của entropy có điều kiện theo thứ tự chuỗi markov cao hơn cho thấy điểm số bất thường trong tương lai của ADS mạng có thể được dự đoán bằng cách sử dụng một vài điểm số trước đó.

câu hỏi này cũng mang lại một mô hình ngẫu nhiên chính xác về điểm số bất thường, do đó có thể được sử dụng để dự đoán ngưỡng.

5.2. Mô hình hóa sự phụ thuộc thời gian trong các điểm dị

thường Ai cũng biết rằng một cấu trúc phụ thuộc thời gian đang phân rã có thể được điều chỉnh chính xác bằng cách sử dụng chuỗi Markov [Merhav et al. 1989]. Do đó, câu hỏi đặt ra ở trên có thể được diễn đạt lại thành: Thứ tự của mô hình chuỗi Markov nên được sử dụng để dự đoán điểm bất thường tiếp theo là gì? Để xác định thứ tự Markovian, chúng tôi sử dụng phép đo dựa trên entropy có điều kiện được đề xuất trong Merhav et al. [1989].

Entropy có điều kiện, H(B|A), của hai biến ngẫu nhiên rời rạc A và B ký tự xác định thông tin còn lại trong B khi A đã biết. Nói cách khác, entropy có điều kiện là "thông tin về B không được cung cấp bởi A." Nếu A và B có mối tương quan cao, hầu hết thông tin về B được truyền đạt qua A và H(B|A) là nhỏ. Mặt khác, nếu pA và pB (tương ứng đại diện cho các hàm khối lượng xác suất của A và B) khác hẳn nhau thì H(B|A) có giá trị cao.1

Để xác định thứ tự của sự hiện diện tương quan trong quy trình chấm điểm ngẫu nhiên của ADS, chúng tôi xác định mô hình ngẫu nhiên dựa trên chuỗi Markov như sau: Đặt điểm số tại thời điểm rời rạc ví dụ n biểu thị việc thực hiện một biến ngẫu nhiên xuất phát từ quy trình ngẫu nhiên Xn. Quá trình này là một chuỗi Markov nếu nó thỏa mãn thuộc tính Markov, được định nghĩa là:

$$\Pr X_n = j \mid X_{n-1} = i, X_{n-2} = i_{n-2}, \ldots, X_0 = i_0 = \Pr X_n = j \mid X_{n-1} = i = p_{j|i}.$$

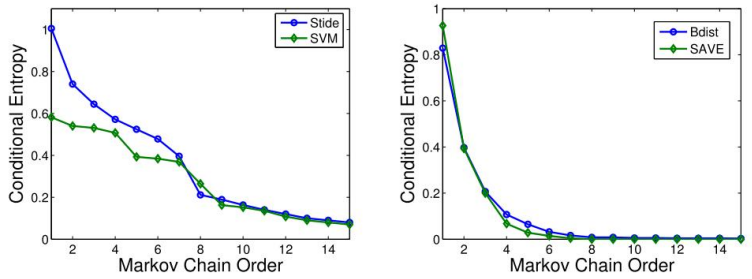
Nói cách khác, xác suất chọn trạng thái tiếp theo chỉ phụ thuộc vào trạng thái hiện tại của chuỗi Markov.

Trong ngữ cảnh hiện tại, chúng ta có thể xác định mô hình chuỗi Markov Xn cho điểm số của ADS bằng cách chia tất cả các giá trị có thể có của điểm số trong nhiều ngăn xếp chồng lên nhau. Sau đó, mỗi bin đại diện cho một trạng thái của chuỗi Markov, trong khi tập hợp tất cả các chỉ số bin ψ là không gian trạng thái của nó. Dựa trên biểu diễn trạng thái này, chúng ta có thể xác định chuỗi Markov bậc 1, X(1) trong đó mỗi ngăn biểu diễn một trạng thái của quá trình ngẫu nhiên. Ma trận xác suất chuyển đổi của chuỗi Markov bậc 1 P(1) có thể được tính bằng cách đếm số lần trạng thái i được theo sau bởi trạng thái j. Kết quả |ψ(1) | biểu đồ có thể được chuẩn hóa để thu được các hàm khối lượng xác suất chuyển tiếp theo trạng thái dưới dạng các hàng của P(1)

Chúng ta có thể tìm xác suất có điều kiện của chuỗi Markov bậc 1 là:

$$H(1) = \sum_{i \in \psi(1)}^{\pi(1)} \sum_{j \in \psi(1)} p(1)_{j|i} \log_2 p(1)_{j|i}, \quad (2)$$

1Trong các trường hợp giới hạn, H(B|A) = 0 khi A = B, trong khi H(B|A) = H(B) khi A và B độc lập.



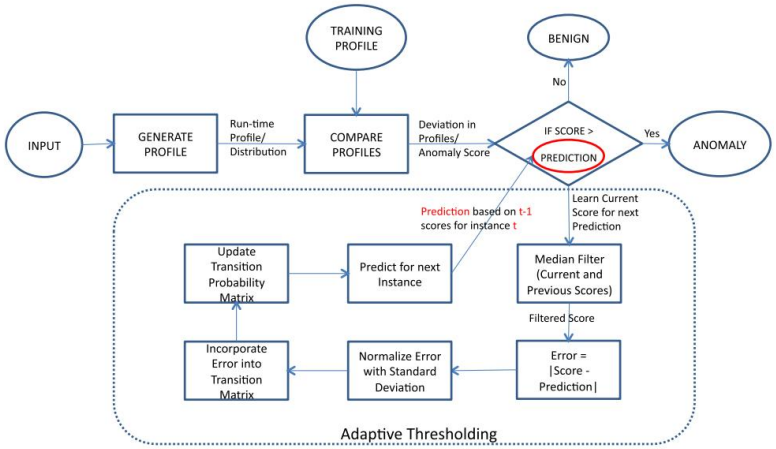
(a) UNM Dataset (b) Nexgin Dataset

Hình 8. Xu hướng suy giảm của entropy có điều kiện theo thứ tự chuỗi markov cao hơn cho thấy các điểm bất thường trong tương lai của ADS lưu trữ có thể được dự đoán bằng cách sử dụng một số điểm trước đó.

trong đó π(1) là xác suất trung bình ở trạng thái i , được tính bằng cách đếm tổng số lần mỗi trạng thái được truy cập và sau đó chuẩn hóa biểu đồ tần số này.

Thước đo H(1) xác định lượng thông tin trung bình còn lại trong điểm bất thường Xn khi nó được dự đoán bằng cách sử dụng điểm Xn 1. Nếu điểm số hiện tại không tương quan cao với điểm số trước Xn 1, thì H(1) sẽ tương đối lớn ngụ ý rằng thông tin về Xn không được cung cấp bởi Xn 1 là cao. Trong trường hợp như vậy, tổng quát hóa cuộc thảo luận ở trên, chúng ta có thể định nghĩa chuỗi Markov bậc cao hơn, X(l) trong đó mỗi trạng thái là một l-bộ < i0, i1, ... , il 1 > đại diện cho các giá trị được thực hiện bởi quá trình ngẫu nhiên trong l trường hợp thời gian qua. Tổng hợp nhiều trường hợp thời gian trong một trạng thái duy nhất cho phép chúng ta thỏa mãn thuộc tính Markov và do đó có thể tính ma trận xác suất chuyển tiếp P(l) bằng cách đếm số lần < i0, i1, ... , il 1 > được theo sau theo trạng thái < i1, ... , il 1, il >. Sau đó, entropy có điều kiện của X(l) được xác định trên ψ(l) có thể được tính bằng phương pháp tương tự như (2). Dễ dàng nhận thấy rằng H(1) ≥ H(2) ≥ ··· ≥ H(l) mỗi điểm , BẰNG bất thường cũ hơn có thể độc lập hoặc cung cấp một số thông tin về điểm hiện tại. Sau đó, số điểm trước đó cần thiết để dự đoán chính xác điểm tiếp theo có thể được xác định bằng cách vẽ đồ thị H(l) là hàm của thứ tự chuỗi Markov, l = 1, 2, ... Thứ tự mà entropy có điều kiện bão hòa xác định tổng số điểm trước đó đã truyền đạt càng nhiều thông tin về điểm hiện tại càng tốt.

Hình 7(a), (b) và (c) hiển thị entropy có điều kiện của các ADS dựa trên mạng trên bộ dữ liệu LBNL, NUST và ISP tương ứng. Có thể quan sát thấy rằng máy dò Entropy cực đại cho thấy xu hướng phân rã theo cấp số nhân trên bộ dữ liệu LBNL cho đến bậc bốn, sau đó phân rã trở nên hơi tuyến tính. Tuy nhiên, trên bộ dữ liệu NUST, sự phân rã theo cấp số nhân là liên tục. Do đó, đối với máy dò Entropy cực đại, bốn giá trị trước đó là đủ để dự đoán điểm số tiếp theo. Tương tự, điểm của TRW có thể được dự đoán bằng cách sử dụng khoảng sáu điểm trước đó trên LBNL và bốn điểm trước đó trên bộ dữ liệu NUST. Như đã mong đợi và đã giải thích trước đó, PHAD thể hiện sự phân rã entropy có điều kiện rất thấp khó có thể nhìn thấy trên thang trục y của Hình 7(a), tuy nhiên có thể thấy một sự phân rã nhỏ trong Hình 7(b). Do đó, điểm số của PHAD có thể được dự đoán chỉ bằng một điểm số trước đó. PAYL cho thấy sự phân rã theo cấp số nhân liên tục trên bộ dữ liệu ISP cho cổng 80 và 21. Mặc dù PAYL cũng phân loại từng gói riêng lẻ như PHAD, nhưng nó cho thấy một số phụ thuộc vì nó thực hiện thuật toán gia tăng trực tuyến, thuật toán này tính đến các gói trước đó để cập nhật cấu hình thông thường. Do đó, điểm số của PAYL có thể được dự đoán bằng cách sử dụng bốn điểm số trước đó. Tương tự, máy dò dựa trên máy chủ cũng cho thấy xu hướng giảm dần tương tự trong Hình 8. STIDE cho thấy mức giảm mạnh nhất, tiếp theo là SVM trong



Hình 9. Đầu vào được sử dụng để tạo cấu hình thời gian chạy, sau đó được so sánh với đào tạo để tạo ra điểm bất thường, sau đó được so sánh với dự đoán (ngưỡng) dựa trên điểm t - 1 cho ví dụ t.

Hình 8(a). Do đó, chúng tôi dự đoán điểm số cho các máy dò STIDE và SVM bằng cách sử dụng tám và chín điểm số trước đó, tương ứng. Hình 8(b) cho thấy xu hướng giảm dần theo cấp số nhân gần như giống nhau đối với cả Bdist và SAVE. Do đó, điểm cho Bdist và SAVE cũng có thể được dự đoán bằng cách sử dụng ba điểm trước đó.

Tại thời điểm này, ngoài việc có một phương pháp chung để xác định số lượng điểm số trước đó cần thiết cho dự đoán điểm số trong tương lai, chúng tôi cũng biết rằng điểm số của ADS có thể được mô hình hóa chính xác bằng cách sử dụng chuỗi Markov rời rạc bậc cao. Trong phần sau, chúng tôi sử dụng mô hình ngẫu nhiên này để phát triển thuật toán theo dõi có thể điều chỉnh chính xác ngưỡng phân loại của ADS.

6. THUẬT TOÁN NGƯỠNG NGƯỠNG THÍ CH ỨNG

Dựa trên kết quả của phần trước, chúng tôi hiện đề xuất một công cụ dự đoán điểm bất thường Marko vian đơn giản và chung chung. Công cụ dự đoán Markovian này về bản chất là một biến thể của thuật toán theo dõi mục tiêu ngẫu nhiên được đề xuất trong [Hollinger et al. 2008]. Tại thời điểm này, điều quan trọng là phải nhắc lại rằng cơ sở lý luận của chúng tôi đối với dự đoán điểm bất thường là điểm dự đoán có thể được sử dụng để ngưỡng điểm trong tương lai phù hợp với các đặc điểm khác nhau. Phần còn lại của phần mô tả thuật toán và so sánh độ chính xác của nó với hai thuật toán dự đoán ngẫu nhiên nổi tiếng.

6.1. Thuật

toán Chúng ta hãy chia nhỏ điểm bất thường của ADS thành k ngăn có kích thước bằng nhau, trong đó k được xác định là sản phẩm phụ của phân tích entropy có điều kiện của phần trước. Cụ thể, bậc Markovian tại đó entropy có điều kiện phân rã bão hòa được chọn làm giá trị của k. Sau đó, kích thước của mỗi thùng được tính bằng cách lấy hiệu của điểm bất thường tối thiểu và tối đa rồi chia cho k. Kích thước của ngăn đầu tiên và ngăn cuối cùng được giữ linh hoạt để phù hợp với mọi điểm bất thường chưa từng thấy trước đây có thể được quan sát thấy trong quá trình vận hành thời gian thực.

Đặt P(n) biểu thị ma trận xác suất chuyển đổi k × k của bộ dự đoán chuỗi Markov tại thời điểm n, trong đó $p(n)_{[i,j]}$ biểu thị một mục tại hàng thứ i và cột thứ j của P(n) . Ngoài ra, đặt r(n) là giá trị thực của điểm ADS được quan sát tại thời điểm n và đặt r^(n) là

Dự đoán Markovian từ trường hợp lần trước. Sau đó, thuật toán cho ngưỡng thích ứng hoạt động như sau:

$$r(n) = \text{medfilt}\{r(n\ T)\ ,r(n\ T+1), \ldots ,r(n)\}, \quad (3)$$

$$\varepsilon(n) = |r(n) \quad \hat{r}(n) \quad (4)$$

$$(n) = \frac{|, \varepsilon(n)}{\sigma r(n)}, \quad (5)$$

$$\bar{p}^{(n+1)}_{r(n)|r(n)} = \beta(n) \times p(n)_{r(n)|r(n)}, \quad (6)$$

$$p^{(n+1)}_{j|r(n)} = \frac{\bar{p}^{(n+1)}_{j|r(n)}}{\sum_{i=1}^{k} \bar{p}^{(n+1)}_{i|r(n)}}, \quad j = 1, \ldots, k, \text{ và } (n+1)$$

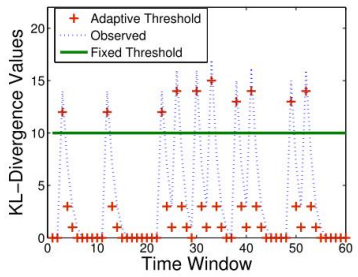
$$\hat{r}(n+1) = \text{tối đa}_{j=1,\ldots,k}\ p^{(n+1)}_{j|r(n)}. \quad (7)$$

Phương trình (3) áp dụng bộ lọc trung bình bậc T trên các giá trị được quan sát để loại bỏ nhiễu ngắn hạn khỏi chúng. Các giá trị được lọc sau đó được sử dụng để dự đoán. Phương trình (4) tính toán sai số dự đoán ε(n) giữa điểm đã lọc được dự đoán và điểm được phục vụ. Sai số được chuẩn hóa bởi σr(n), là độ lệch chuẩn của hàng P(n) tương ứng với r(n) . Sau đó, tại mỗi thời điểm bước n, Eq. (6) cung cấp lỗi chuẩn hóa bình thường β(n) trở lại P(n) để thích ứng và tìm hiểu các mẫu lưu lượng truy cập hoặc máy chủ lưu trữ khác nhau. Sử dụng phản hồi lỗi này, trọng số của một giá trị được quan sát r(n) được tăng theo tỷ lệ bằng cách sử dụng tham số β làm trọng số. Do đó, lỗi cao hơn có nghĩa là xác suất của các trạng thái này tăng theo tỷ lệ thuận và các giá trị dự đoán cho trường hợp thời gian tiếp theo sẽ có khả năng trôi xa khỏi trạng thái hiện tại. Hàng cập nhật của ma trận xác suất chuyển đổi được chuẩn hóa lại để có được xác suất cập tiếp theo, hàm khối lượng nhật . Cuối cùng, phương trình. (7) dự đoán điểm bất thường , r̂(n+1) cho trạng thái r(n) là trạng thái có xác suất cao nhất trong ma trận xác suất chuyển đổi được cập nhật. Điểm bất thường được dự đoán này được sử dụng làm ngưỡng thích ứng cho trường hợp thời gian n + 1.

Hình 9 cho thấy hoạt động cấp cao của thuật toán tạo ngưỡng thích ứng được đề xuất với các ADS hiện có, hoạt động theo cách ngưỡng. Có thể thấy rằng nó coi ADS như một hộp đen và lấy các điểm bất thường do ADS tạo ra làm đầu vào. Dựa trên đầu vào, nó dự đoán điểm bất thường cho khoảng thời gian tiếp theo, sau đó được sử dụng làm ngưỡng. ADS lấy lưu lượng truy cập hoặc dữ liệu máy chủ lưu trữ (tại thời gian chạy) làm đầu vào được sử dụng để tạo hồ sơ/phân phối thời gian chạy. Hồ sơ thời gian chạy/phân phối này sau đó được so sánh với hồ sơ đào tạo/dsitribution được tạo trước đó và sự khác biệt giữa hai hồ sơ này được tính toán, đó là điểm bất thường. Các ADS khác nhau sử dụng các phương pháp khác nhau để tạo và so sánh các bản phân phối này. Nói chung, điểm số bất thường này được so sánh với một ngưỡng để phân loại đầu vào là bất thường (nếu lớn hơn ngưỡng) hoặc không đáng kể (nếu nhỏ hơn ngưỡng). Tuy nhiên, ngưỡng thích ứng đặt ngưỡng được dự đoán dựa trên t 1 phiên bản trước đó cho phiên bản t hiện tại. Do đó, ngưỡng sẽ được dự đoán cho mọi phiên bản bằng cách sử dụng điểm số của các phiên bản trước đó.
Vì các ADS truyền thống hoạt động theo phương pháp ngưỡng, ngưỡng thích ứng có thể dễ dàng được cắm vào như trong Hình 9.

6.2. Độ chính xác dự đoán của thuật toán ngưỡng thích ứng

Để đánh giá độ chính xác, chúng tôi so sánh độ chính xác dự đoán của thuật toán được đề xuất với hai công cụ dự đoán nổi tiếng, đó là bộ lọc Kalman và Holt-Winters [Trees 2001]. Hình 3(a) và (b) cho thấy độ chính xác của ba yếu tố dự báo (Markovian,



Hình 10. Ngưỡng cố định không thể đưa ra cảnh báo về phân kỳ KL (Entropy tối đa), tuy nhiên, ngưỡng thích ứng đã điều chỉnh và đưa ra cảnh báo trong cửa sổ dị thường.

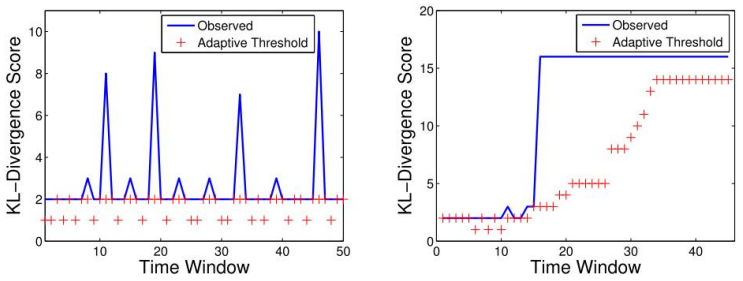
Kalman, Holt-Winters) trong việc theo dõi các xu hướng đầu vào (lưu lượng truy cập và cuộc gọi hệ thống) được quan sát thấy trong bộ dữ liệu LBNL và UNM. Có thể thấy trong Hình 3 rằng, trong khi cả ba bộ dự đoán có thể theo dõi các phép đo thời gian thực, thì bộ dự đoán Markovian theo dõi các phép đo này chính xác hơn nhiều so với bộ lọc Kalman và bộ dự đoán Holt-Winters.

Là một ví dụ chứng minh khái niệm về các lợi thế về độ chính xác có thể được cung cấp bởi ngưỡng thích ứng, Hình 10 cho thấy các ngưỡng do Markovian dự đoán trong cửa sổ thời gian LBNL dị thường cho máy dò Entropy cực đại; ngưỡng trong trường hợp này là phân kỳ KL của một loại gói tin đã bị nhiễu do tack. Ngoài ra, các giá trị phân kỳ KL (trục y) biểu thị các thùng có kích thước bằng nhau, mỗi thùng chứa 2 giá trị, nghĩa là các giá trị [0,1] thuộc về thùng 0. Xin lưu ý rằng với mục đích quan sát độ chính xác của dự đoán, chúng tôi đã vô hiệu hóa biểu thức. (3) bằng cách đặt T = 0 và phương trình. (5).
Có thể thấy rõ rằng công cụ dự đoán Markovian ước tính các giá trị phân kỳ KL rất khác nhau với độ chính xác đáng kể.

Cũng rất thú vị khi lưu ý rằng trong Hình 10, việc chọn một ngưỡng cố định có thể cho phép một số điểm bất thường không bị phát hiện, đặc biệt là các điểm bất thường không gây ra nhiễu loạn đáng kể trong lưu lượng mạng thực tế. Chẳng hạn, trong đầu ra 60 giây được hiển thị trong Hình 10, chỉ có 10 giá trị trong số này vượt qua ngưỡng cố định. Trong thử nghiệm này, thuật toán Entropy cực đại đã phát hiện ra sự bất thường nếu 12 giá trị trở lên trong khoảng thời gian 60 giây vượt quá ngưỡng cố định. Do đó, sự bất thường này sẽ không được phát hiện bởi một ngưỡng cố định. Mặt khác, ngưỡng thích ứng dự đoán chính xác sự phân kỳ KL trong cửa sổ tiếp theo và sự phân kỳ (nhiễu loạn) được quan sát vượt quá ngưỡng này hơn 20 lần trong cửa sổ 60 giây, do đó cho phép máy dò Entropy cực đại đánh dấu sự bất thường. Cuối cùng, có thể quan sát từ Hình 10, rằng trong vài giây, các giá trị KL giảm xuống 0. Các giá trị thấp này tạo cho kẻ tấn công xảo quyệt đòn bẩy để đưa ra lưu lượng độc hại không vượt quá ngưỡng cố định [0, 10]. Tuy nhiên, ngưỡng thích ứng sẽ ngay lập tức tìm hiểu sự thay đổi và đặt ngưỡng thành 0, do đó đảm bảo rằng không có chỗ cho một cuộc tấn công bất chước như vậy.

6.3. Tính ổn định của thuật toán ngưỡng thích ứng Chúng

tôi đã chứng minh rằng bộ dự đoán được đề xuất cung cấp các ước tính tốt hơn đáng kể so với bộ lọc Kalman và bộ dự đoán Holt-Winters. Ngoài việc học hiệu quả hành vi tạm thời của điểm số bất thường, chúng tôi cũng muốn người dự đoán được đề xuất thể hiện hành vi ổn định khi quan sát thấy những thay đổi lẻ tẻ trong các cuộc gọi hệ thống/lưu lượng truy cập. Các nghiên cứu trước đây về lưu lượng truy cập FTP, TELNET, WWW, NNTP và SMTP đã chỉ ra rằng phương sai lưu lượng truy cập đáng kể (độ bùng nổ) xuất hiện trên một dải thời gian rộng [Crovella và Bestavros 1997]. Do những thay đổi này trong các đặc điểm đầu vào, những thay đổi đột ngột về điểm số bất thường được quan sát thấy và các dự đoán về điểm số bất thường tiếp theo của chúng dao động. Mặc dù những thay đổi này có tính chất rời rạc và ngắn hạn, nhưng chúng



(a) Stability on NUST Dataset (b) Evasion on NUST Dataset

Hình 11. Ngưỡng thích ứng không học các dao động ngắn hạn, tuy nhiên, nó dần dần học các thay đổi dài hạn do đó mang lại sự ổn định và khả năng tránh né mạnh mẽ.

có thể làm mất ổn định một yếu tố dự đoán và do đó làm giảm độ chính xác dự đoán của nó. Những thay đổi ngắn hạn này nên được coi là nhiễu vì những thay đổi này không phản ánh những sai lệch lâu dài hoặc lâu dài trong hành vi đầu vào.

Để làm cho bộ dự đoán không nhạy cảm với các nhiễu loạn lẻ tẻ và ngắn hạn này, chúng tôi áp dụng bộ lọc trung vị (Phương trình (3)) trên các điểm bất thường quan sát được trước khi dự đoán điểm tiếp theo. Bộ lọc trung vị lưu trữ T giá trị trước đó của đầu vào và ở mỗi bước xuất ra trung vị của các giá trị được lưu trữ. Điều này dẫn đến việc loại bỏ các đột biến tần số cao khỏi dữ liệu đầu vào. Cần lưu ý rằng giá trị của T đại diện cho giới hạn trên thô trong khoảng thời gian tối đa của một đợt tăng đột biến ngắn hạn bất thường. Chúng tôi đặt T bằng với thứ tự của mô hình chuỗi Markov được sử dụng để dự đoán. Nếu thay đổi tồn tại lâu hơn T quan sát, thì nó được coi là thay đổi dài hạn trong hành vi của đầu vào.

Hình 10 cho thấy một ví dụ dự đoán cho bộ phát hiện Entropy cực đại không có bộ lọc trung vị (Phương trình (3)). Có thể thấy từ Hình 10 rằng khi có sự khác biệt ngắn hạn trong đầu vào, giá trị ngưỡng dự đoán tương ứng cũng thay đổi. Mặt khác, khi sử dụng bộ lọc trung vị, các điểm bất thường tăng đột biến trong thời gian ngắn sẽ bị bỏ qua, dẫn đến thuật toán dự đoán ổn định hơn. Hình (11)(a) cho thấy điểm bất thường quan sát được trong thời gian thực đối với máy dò Entropy cực đại trên bộ dữ liệu NUST. Có thể quan sát thấy rằng bằng cách sử dụng bộ lọc trung bình, các đợt bùng phát ngắn hạn lẻ tẻ đã được lọc ra và cường độ dao động trong giá trị ngưỡng dự đoán được giảm đáng kể.

6.4. Mạnh mẽ chống lại các cuộc tấn công trốn tránh

Các cuộc tấn công trốn tránh có thể được gắn vào các thuật toán ngưỡng thích ứng được đề xuất bởi kẻ tấn công có thể tạo lưu lượng truy cập mạng hoặc các cuộc gọi hệ thống một cách thông minh làm thay đổi hành vi đầu vào của ADS. Điều này sẽ cho phép kẻ tấn công khởi động một cuộc tấn công mà không bị phát hiện bằng cách thay đổi ấn tượng về lưu lượng lành tính của ADS để phù hợp hơn với ấn tượng của cuộc tấn công; do đó, cuộc tấn công sẽ được che đậy bởi kỳ vọng của ADS về lưu lượng lành tính và sẽ được đảm bảo giảm xuống dưới ngưỡng. Để cung cấp khả năng phục hồi chống lại một cuộc tấn công né tránh như vậy, thuật toán ngưỡng thích ứng không nên điều chỉnh ngay lập tức với đầu vào. Do đó, sự mạnh mẽ chống trốn tránh là về thời gian học tập. Vì ngưỡng không cố định và nó thích nghi, mối quan tâm ở đây là tốc độ thích ứng và ổn định của nó. Nếu nó thích ứng với một vài mẫu (tức là trong khoảng thời gian ngắn), thì kẻ tấn công xảo quyệt có thể thay đổi hành vi bình thường bằng cách tạo ra một vài mẫu trong thời gian ngắn và sau đó khởi động một cuộc tấn công thực sự để không bị phát hiện. Tuy nhiên, thuật toán ngưỡng thích ứng được thiết kế để học chậm dựa trên sự khác biệt về điểm số được quan sát và dự đoán đối với độ lệch chuẩn. Tiêu chuẩn này

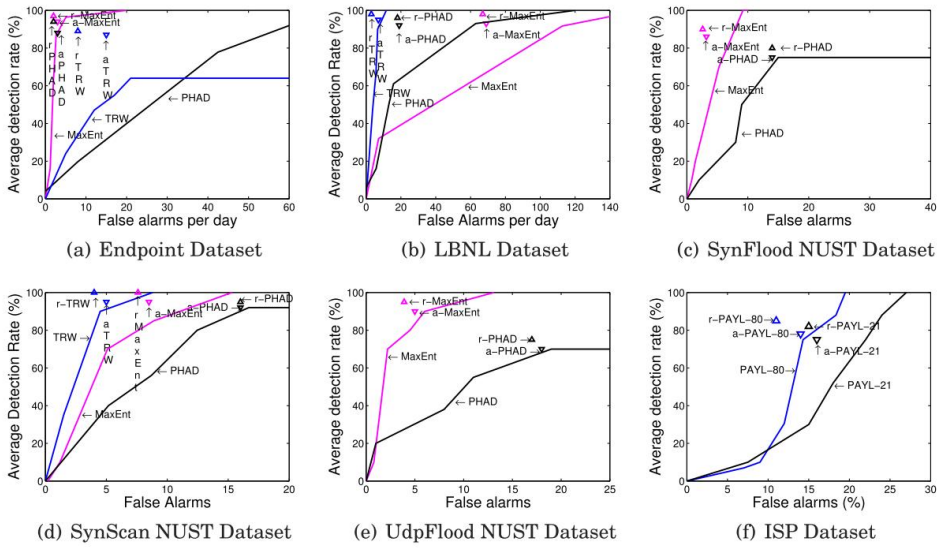
sai lệch khiến nó học chậm và do đó, kẻ tấn công phải đợi lâu hơn để không bị phát hiện bằng cách làm cho ADS học theo xu hướng mới.

Ví dụ: nếu kẻ tấn công muốn khởi động một cuộc tấn công Từ chối Dịch vụ (DoS), anh ta có thể gửi các gói lành tính hơi tuân theo các đặc điểm của cuộc tấn công do ADS xác định (chẳng hạn như phân phối cụ thể cho trình phát hiện Entropy Tối đa). ADS sẽ thấy sự thay đổi về đặc điểm lưu lượng truy cập và với điều kiện là lưu lượng truy cập này không vượt quá ngưỡng, ADS sẽ tăng ngưỡng của nó một cách thích ứng để quản lý bản chất đã thay đổi của lưu lượng truy cập bình thường. Sau một thời gian, kẻ tấn công một lần nữa có thể thao túng các đặc điểm lưu lượng lành tính để gần với lưu lượng tấn công mong muốn hơn. Một cách ổn định nhưng chắc chắn, ADS sẽ thích ứng với điều này và một khi ngưỡng phát hiện tấn công DoS đủ thay đổi, kẻ tấn công cuối cùng có thể khởi chạy cuộc tấn công DoS và duy trì ở dưới mức ngưỡng.

Để tránh các cuộc tấn công như vậy, thuật toán ngưỡng thích ứng được đề xuất sẽ bình thường hóa lỗi (sự khác biệt giữa điểm ADS được quan sát và điểm dự đoán) có liên quan đến độ lệch chuẩn của lỗi, mất nhiều thời gian hơn để tìm hiểu sự thay đổi trong hành vi đầu vào. Phương trình (5) trong thuật toán cho phép thuật toán lão hóa ngưỡng thích ứng mất nhiều thời gian hơn để tìm hiểu những thay đổi đó. So sánh Hình 11(b) với Hình 10 cho thấy khả năng tránh né mạnh mẽ của thuật toán tạo ngưỡng được đề xuất. Trong thử nghiệm của chúng tôi, vì tính mạnh mẽ chống trốn tránh là một đặc điểm của ngưỡng thích ứng, để chứng minh khái niệm, chúng tôi đã sử dụng trình phát hiện Entropy cực đại để xây dựng cấu hình dựa trên giao thức và cổng đích. Nó tính toán phân kỳ KL cho cấu hình bình thường và thời gian chạy, do đó so sánh nó với một ngưỡng. Để tăng sự khác biệt, chúng tôi đã khởi chạy TCP SYN (100, 1000 gói/giây) trên các cổng và IP đích khác nhau. Có thể thấy trong Hình 11(b) rằng hành vi của lưu lượng đã thay đổi được biểu thị bằng một đường liền nét. Ngay sau khi hành vi được thay đổi bằng cách sử dụng tốc độ cao trong các lần quét cổng TCP SYN, cuộc tấn công lũ lụt UDP (0,1, 1, 10 pkts/giây) được nhắm mục tiêu với tốc độ thấp đã được khởi chạy trên hai máy chủ từ xa để vượt qua sự phát hiện. Có thể quan sát trong Hình 11(b) rằng thuật toán dần dần thích ứng với thay đổi hành vi trong đầu vào, do đó đưa ra cảnh báo về cuộc tấn công UDPFlood. Điều này cho phép hệ thống đưa ra cảnh báo trước khi trở thành nạn nhân của các cuộc tấn công trốn tránh và kích hoạt chính sách phù hợp để xử lý một cuộc tấn công trốn tránh như vậy. Để trốn tránh, kẻ tấn công phải thay đổi hành vi lưu lượng truy cập trong một thời gian dài cho đến khi ngưỡng thích ứng điều chỉnh hành vi được quan sát và sau đó khởi chạy một cuộc tấn công thực sự để vượt qua sự phát hiện. Tuy nhiên, chúng tôi tin rằng việc thay đổi hành vi trong một thời gian dài sẽ khiến nó dễ bị phát hiện hơn.

7. ĐÁNH GIÁ

Thuật toán ngưỡng thích ứng được đề xuất là chung vì nó không dựa vào các tính năng và nguyên tắc phát hiện của các ADS cơ bản và có thể dễ dàng cắm vào bất kỳ ADS hiện có nào hoạt động theo nguyên tắc ngưỡng. Bây giờ chúng tôi đánh giá thuật toán ngưỡng được đề xuất bằng cách kết hợp nó vào tám ADS. Chúng tôi hy vọng thuật toán ngưỡng thích ứng sẽ chọn các điểm ngưỡng tốt hơn đáng kể trên mặt phẳng ROC so với giá trị ngưỡng cố định. Chúng tôi trình bày kết quả đánh giá của chúng tôi đối với hai phiên bản của thuật toán ngưỡng thích ứng: (1) chỉ mô-đun ngưỡng thích ứng không có độ ổn định và khả năng trốn tránh (những kết quả này được gắn nhãn là tên A-ADS); và (2) kết quả với các mô-đun độ ổn định và khả năng trốn tránh (được gắn nhãn là tên r-ADS). Ngoài những cải tiến về hiệu suất đạt được so với các ADS hiện có, việc so sánh độ chính xác với hai kỹ thuật đối ứng để trôi khái niệm, đó là ECSMiner [Masud et al. 2011] và Công cụ khai thác đa lớp (MCM) [Masud et al. 2010], được hiển thị. Cuối cùng, chúng tôi cũng chỉ ra rằng độ phức tạp bổ sung do ngưỡng thích ứng đưa ra là không đáng kể so với độ phức tạp của ADS không có mô-đun ngưỡng thích ứng.



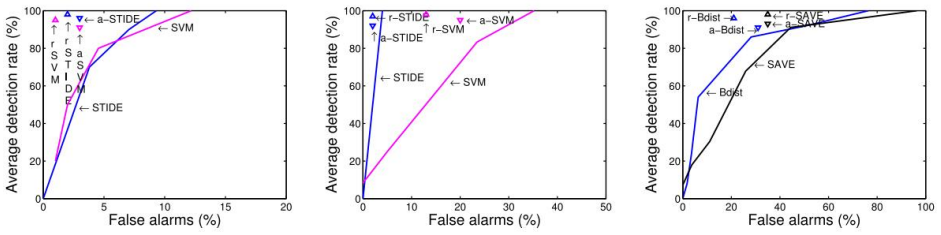
Hình 12. Các cải tiến về độ chính xác đạt được bằng cách đặt ngưỡng thích ứng trên các ADS dựa trên mạng.

7.1. Đánh giá độ chính xác Trước

khi thảo luận về những cải tiến về độ chính xác đã đạt được, trước tiên chúng tôi giải thích việc triển khai thuật toán được đề xuất với các máy dò hiện có được sử dụng trong công việc của chúng tôi. Đối với Entropy tối đa, chúng tôi đã cắm bộ phát hiện được đề xuất để dự đoán giá trị phân kỳ KL của phiên bản tiếp theo (vì giá trị KL được sử dụng làm tham số điều chỉnh). Vì PHAD tính toán điểm gói dựa trên các giá trị trường tiêu đề, chúng tôi sử dụng mô-đun cũ ngưỡng thích ứng để tìm hiểu và dự đoán các giá trị điểm gói trong tương lai. TRW hoạt động dựa trên tỷ lệ khả năng để phân loại nguồn là máy quét. Chúng tôi đã sử dụng thuật toán của mình để dự đoán giá trị tỷ lệ khả năng tiếp theo. Cụ thể hơn, chúng tôi đã áp dụng nó ở giới hạn trên trong thuật toán TRW, dẫn đến giả thuyết rằng nguồn là một máy quét. PAYL hoạt động dựa trên việc tính toán khoảng cách Mahalanobis giữa cấu hình thời gian chạy và cấu hình bình thường, do đó, ngưỡng thích ứng được cắm vào để dự đoán khoảng cách của gói tiếp theo so với cấu hình bình thường.

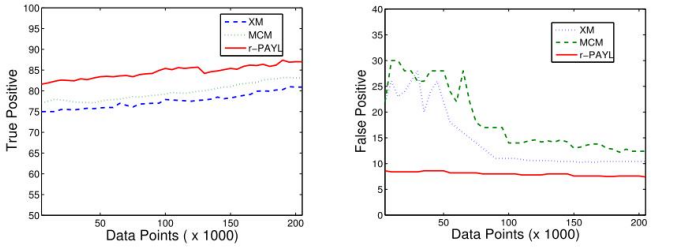
Tương tự như vậy đối với STIDE, thuật toán đề xuất được sử dụng để dự đoán tỷ lệ phần trăm khớp sai giữa các trình tự từ cấu hình lành tính và các trình tự từ cấu hình độc hại. Trong trường hợp của SVM, thuật toán được đề xuất được sử dụng để xác định tổng số hạt nhân cho tập hợp tần suất tiếp theo của các cuộc gọi hệ thống tới bộ dò. Cuối cùng, đối với SAVE và Bdist, giá trị tương quan kết hợp và giá trị ngưỡng khoảng cách tiến hóa được dự đoán tương ứng. Cả hai giá trị này đều được dự đoán bất cứ khi nào một lệnh gọi API được thêm vào các chuỗi hiện có.

Hình 12 cho thấy sự so sánh độ chính xác dựa trên ROC của các máy dò Entropy, TRW, PHAD và PAYL tối đa có và không có ngưỡng thích ứng. Đối với tập dữ liệu Điểm cuối, Hình 12(a) cho thấy a-PHAD giảm đáng kể cảnh báo sai của PHAD bằng cách tìm hiểu những thay đổi hợp pháp trong hành vi lưu lượng mạng. Tương tự, r PHAD cải thiện hơn nữa độ chính xác của a-PHAD bằng cách giảm 10% cảnh báo sai và cải thiện 4% tỷ lệ phát hiện. Các kết quả tương tự về mặt định tính có thể được quan sát đối với a-TRW và r-TRW. Chúng tôi không nhận được những cải tiến đáng kể trong trường hợp trình phát hiện a-MaxEnt trên tập dữ liệu Điểm cuối vì thuật toán ban đầu cung cấp độ chính xác cao. Tuy nhiên, những cải tiến cận biên có thể được quan sát thấy trong trường hợp của r-MaxEnt.



(a) UNM Dataset (b) MIT-LL Dataset (c) Nexgin Dataset

Hình 13. Cải thiện độ chính xác đạt được bằng cách đặt ngưỡng thích ứng đối với ADS dựa trên máy chủ.



(a) Detection on ISP Dataset (b) False Positive on ISP Dataset

Hình 14. So sánh độ chính xác của XM, MCM và Ngưỡng thích ứng trên Bộ dữ liệu ISP.

Hình 12(b) cho thấy sự so sánh độ chính xác trên bộ dữ liệu LBNL. Có thể thấy rằng MaxEnt không thể duy trì mức tăng hiệu suất so với bộ dữ liệu Điểm cuối do số lượng lớn cảnh báo sai được đưa ra bởi các biến thể lưu lượng thất thường ở bộ định tuyến biên. Trong trường hợp của a-MaxEnt, các cải tiến đáng kể về độ chính xác của phát hiện và giảm cảnh báo sai được ghi nhận do khả năng của thuật toán học các biến thể lưu lượng. Tương tự, r-MaxEnt tiếp tục giảm thêm 2% cảnh báo sai và cải thiện một chút về độ chính xác của phát hiện. Các kết quả tương tự về mặt định tính có thể được quan sát đối với a-PHAD và r-PHAD. Các cải tiến cận biên đã được quan sát đối với a- và r-TRW do tỷ lệ phát hiện cao hiện có của TRW.

Từ bộ dữ liệu NUST, chúng tôi báo cáo kết quả cho SynFlood, SynScan và UDPFlood. Có thể quan sát thấy từ Hình 12(c), Hình 12(d) và Hình 12(e) rằng các cải tiến độ chính xác đáng kể và nhất quán đạt được nhờ các ADS thích ứng so với các đối tác ban đầu của chúng. Cần lưu ý rằng các kết quả cho TRW trong các cuộc tấn công SynFlood và UDPFlood không được cung cấp vì TRW sử dụng các gói trả lời để phát hiện và các gói trả lời cho SynFlood và UDPFlood không tồn tại. Một quan sát thú vị khác là r-PHAD cung cấp các cải tiến về hiệu suất so với a-PHAD vì r-PHAD không tìm hiểu các biến thể lưu lượng truy cập ngay lập tức và do đó cải thiện độ chính xác phát hiện thêm 7%.

Hình 12(f) cho thấy sự so sánh độ chính xác của bộ phát hiện PAYL trên bộ dữ liệu ISP cho cổng 80 và 21 được ký hiệu lần lượt là PAYL-80 và PAYL-21. Có thể thấy rằng các cải tiến nhất quán được nhìn thấy đối với ngưỡng thích ứng so với thuật toán ban đầu. Nó cho thấy rằng r-PAYL cải thiện tỷ lệ phát hiện trong cả hai trường hợp, tuy nhiên, việc giảm đáng kể cảnh báo sai chỉ được quan sát thấy trong trường hợp cổng 80.

Hình 13 cho thấy sự so sánh dựa trên ROC của các bộ dò STIDE, SVM, Bdist và SAVE có và không có phiên bản ngưỡng thích ứng. Có thể thấy rằng a-STIDE và r-STIDE giúp cải thiện tới 30% độ chính xác trong phát hiện và giảm tới 50% tỷ lệ cảnh báo sai trên tập dữ liệu UNM. Chứng minh tương tự về mặt chất lượng đạt được bằng cách sử dụng a-SVM và r-SVM trên cả bộ dữ liệu UNM và MIT LL.

SAVE không thể cung cấp độ chính xác có thể chấp nhận được trên bộ dữ liệu Nexgin, tuy nhiên, r-SAVE đã cải thiện tới 15% tỷ lệ phát hiện và giảm đáng kể các cảnh báo sai. Cuối cùng, một số cải thiện về tỷ lệ phát hiện đã được quan sát đối với r-Bdist so với bản gốc và a-Bdist.

Hình 14 cho thấy sự so sánh độ chính xác của ngưỡng thích ứng với hai kỹ thuật đối ứng được đề xuất gần đây, đó là ECSMiner (XM) [Masud et al. 2011] và Công cụ khai thác đa lớp (MCM) [Masud et al. 2010]. Máy dò PAYL đã được sử dụng làm công cụ học cơ sở cho cả hai kỹ thuật. Chúng tôi đã sử dụng các tham số cấu hình giống như trong Masud et al. [2011]. Số lượng điểm giả trên mỗi bộ phân loại được đặt thành 50. Kích thước của bộ đồng phục là 6 đã được sử dụng. Kích thước khối được giữ ở mức 2000. Chúng tôi đã bỏ qua độ chính xác phát hiện lớp mới vì điều đó nằm ngoài phạm vi công việc của chúng tôi. Tuy nhiên, chúng tôi chỉ tập trung vào độ chính xác của việc phát hiện tấn công và xác định sai đối với lưu lượng truy cập bình thường. Bộ dữ liệu ISP cho cổng 80 đã được sử dụng. Có thể thấy rõ rằng r-PAYL vượt trội hơn cả hai kỹ thuật vì góc ngẫu nhiên mang lại giá trị trong việc xử lý các hành vi lẻ tẻ so với cơ chế xác định.

7.2. Đánh giá và triển khai độ phức tạp Chúng

tôi đã đo độ phức tạp trong thời gian chạy thuật toán và yêu cầu bộ nhớ bằng công cụ hprof [Heap Profiler] trên máy Intel lõi kép 2,2 GHz. Độ phức tạp được đo lường bằng cách chạy các thuật toán trên bộ dữ liệu điểm cuối, ISP và UNM.
Độ phức tạp đối với các mô-đun ngưỡng thích ứng kế thừa (a) và mạnh mẽ (r) trên dữ liệu lưu lượng truy cập của điểm cuối lần lượt là khoảng 60 và 80 mili giây, ít hơn hai và năm bậc độ lớn so với thời gian cần thiết để thực hiện Entropy cực đại và TRW /PHAD thuật toán phát hiện bất thường, tương ứng. Ví dụ: độ phức tạp được quan sát cho Entropy tối đa là 22,79 giây. Tuy nhiên, khi được sử dụng với kỹ thuật ngưỡng thích ứng mạnh mẽ, độ phức tạp hóa ra là 22,87 giây (tức là 80 mili giây trên đầu). Tương tự, các yêu cầu về bộ nhớ dữ liệu của thuật toán tạo ngưỡng thích ứng cũng không đáng kể (vài trăm KB, xấp xỉ 400) đối với bộ nhớ được sử dụng bởi các thuật toán phát hiện bất thường; ba bậc độ lớn cho Entropy cực đại và hai bậc độ lớn cho TRW và PHAD. Các kết quả về độ phức tạp tương tự (xấp xỉ hai bậc độ lớn trong thời gian chạy và một bậc độ lớn trong bộ nhớ dữ liệu) đã được quan sát đối với các máy dò dựa trên máy chủ. Độ phức tạp của máy dò PAYL được quan sát là 21,31 giây. Tuy nhiên, nó là 21,39 giây khi được sử dụng với ngưỡng thích ứng.

Do đó, ngưỡng thích ứng đã giới thiệu chi phí hoạt động là 80 mili giây trên máy 2,2 GHz. XM và MCM quan sát thấy độ phức tạp lần lượt là 28,35 và 26,67 giây. Tương tự, yêu cầu bộ nhớ cao hơn nhiều đối với chúng, nghĩa là, vài MB so với ngưỡng thích ứng yêu cầu vài KB.

Việc triển khai ngưỡng thích ứng với các ADS hiện tại khá đơn giản.
ADS định lượng sự khác biệt trong cấu hình bình thường và thời gian chạy. Giá trị đo lường có thể định lượng này được so sánh với một ngưỡng cố định trong bộ phân loại. Ngưỡng thích ứng có thể được triển khai theo hai cách: (1) thay thế số ngưỡng cố định bằng một biến có giá trị có thể được đặt bởi mô-đun ngưỡng thích ứng, (2) thay thế số ngưỡng cố định bằng một lệnh gọi phương thức trả về số ngưỡng thích ứng mới .
Lợi ích của cách tiếp cận thứ hai là giá trị được quan sát trong thời gian chạy có thể được gửi đến ngưỡng thích ứng làm đối số cho việc học để dự đoán ngưỡng cũ mới được trả về.

7.3. Hạn chế

— Ngưỡng thích ứng chỉ có thể được áp dụng cho các bộ phát hiện thời gian thực hoạt động theo cách ngưỡng, nghĩa là các bộ phát hiện so sánh cấu hình đã học với cấu hình thời gian chạy bằng cách sử dụng thước đo có thể định lượng theo ngưỡng để phân loại. Vì thế,

nó không thể được áp dụng cho hầu hết các trình phát hiện dựa trên máy học hoặc phân cụm như phân tích tệp (máy học) và thuật toán phân loại lưu lượng dựa trên quy tắc.

– Một số điểm bất thường có thể không bị phát hiện nếu lưu lượng tấn công dần dần thao túng hành vi, tức là các cuộc tấn công tốc độ thấp tạo ra lưu lượng rất ít so với lưu lượng thông thường. Một biện pháp đối phó đơn giản cho vấn đề này là đặt độ dài cửa sổ bộ lọc trung bình thích hợp. Kích thước cửa sổ càng lớn thì tốc độ thích ứng càng chậm, nghĩa là nó có xu hướng hoạt động giống như ngưỡng cố định. Trong các thử nghiệm của mình, chúng tôi đã thử nghiệm cách tiếp cận đối với các cuộc tấn công tốc độ thấp cũng như trong bộ dữ liệu NUST, nghĩa là, thấp tới 0,1 pkts/giây.

8. KẾT LUẬN

Chúng tôi đã chỉ ra rằng việc đánh giá độ chính xác dựa trên ROC bằng cách sử dụng các ngưỡng cố định không nhất thiết phải đại diện cho độ chính xác thực tế mà một ADS có thể đạt được.
Hơn nữa, chúng tôi đã trình bày phần mở rộng của thuật toán trước đó [Ali et al. 2009] và chứng minh những cải tiến hiệu suất hơn nữa trên ba số liệu quan trọng: độ chính xác, độ ổn định và khả năng trốn tránh mạnh mẽ. Chúng tôi đã đề xuất một thuật toán điều chỉnh ngưỡng chung: (1) cho phép ADS đạt được các điểm có độ chính xác cao trên mặt phẳng ROC; (2) có thể dễ dàng đưa vào các ADS hiện có; (3) giảm nhu cầu về cấu hình cũ đập của con người trong ADS; và (4) có yêu cầu về bộ nhớ và độ phức tạp trong thời gian chạy rất thấp.

RUỘT THỪA

A. BỘ SỐ LIỆU SỬ DỤNG ĐỂ ĐÁNH GIÁ

Trong phần này, chúng tôi mô tả lưu lượng truy cập mạng và bộ dữ liệu trình tự lệnh gọi/API hệ thống được sử dụng để đánh giá trong công việc này.

A.1. Bộ dữ liệu lưu lượng mạng

Chúng tôi sử dụng bốn bộ dữ liệu lưu lượng đã được thu thập độc lập tại các điểm triển khai khác nhau. Tất cả các bộ dữ liệu được gắn nhãn và có sẵn công khai tại [Bộ dữ liệu LBNL], [WisNet ADS] và [Bộ dữ liệu NUST] ngoại trừ bộ dữ liệu ISP.

A.1.1. Bộ dữ liệu LBNL. Bộ dữ liệu này được thu thập tại hai địa điểm mạng quốc tế tại Phòng thí nghiệm quốc gia Lawrence Berkeley (LBNL), Hoa Kỳ. Các ứng dụng chính trong lưu lượng bên trong và bên ngoài là Web (HTTP), Email và Dịch vụ tên. Một số ứng dụng khác như Dịch vụ Windows, Dịch vụ tệp mạng và Sao lưu cũng đang được sử dụng bởi các máy chủ nội bộ. Lưu lượng độc hại chủ yếu bao gồm các yêu cầu TCP SYN đến không thành công được nhắm mục tiêu tới các máy chủ LBNL; xem Pang et al. [2005] để biết chi tiết. Một số thống kê thích hợp của tập dữ liệu LBNL được đưa ra trong Bảng I. Lưu ý rằng tốc độ tấn công thấp hơn đáng kể so với tốc độ lưu lượng truy cập nền. Do đó, các cuộc tấn công này có thể được coi là tốc độ thấp so với tốc độ lưu lượng truy cập nền. Chúng tôi đã lọc lưu lượng truy cập cục bộ từ tập dữ liệu.

A.1.2. Bộ dữ liệu điểm cuối. Tập dữ liệu này bao gồm lưu lượng truy cập cấp phiên được thu thập tại 13 điểm cuối công việc trên mạng. Người dùng của những điểm cuối này bao gồm người dùng gia đình, sinh viên nghiên cứu và nhân viên kỹ thuật/hành chính. Các điểm cuối đang chạy các loại ứng dụng khác nhau, bao gồm phần mềm chia sẻ tệp ngang hàng, ứng dụng đa phương tiện trực tuyến, trò chơi mạng, máy khách SQL/SAS, v.v. Thống kê về điểm cuối có tỷ lệ lưu lượng truy cập lành tính cao nhất và thấp nhất được trình bày trong Bảng II. Lưu lượng tấn công trong bộ dữ liệu này chủ yếu bao gồm các lần quét cổng đi; xem Ashfaq và cộng sự. [2008] để biết chi tiết. Lưu lượng tấn công được tạo bằng cách sử dụng phần mềm độc hại sau: Zotob.G, Forbot-FU, Sdbot-AFR, Dloader-NY, SoBig.E@mm, MyDoom.A@mm, Blaster, Rbot-AQJ và RBOT.CCC [Symantec Security]. Bảng III cho thấy số liệu thống kê về các loại sâu có tốc độ quét cao nhất và thấp nhất. Để hoàn thiện, chúng tôi cũng

Bảng I. Thông tin lưu lượng cơ bản cho Tập dữ liệu LBNL Ngày Trung

bình từ xa LBNL. Nền trung bình Tỷ lệ Pkt tấn công máy chủ (/giây)				
		chủ nhà		Tỷ lệ gói (/giây)
4/10/04 4.767 4.342 15/12/04			8,47 0,41	
5.761 10.478 16/12/04 5.210 7.138			3,5	0,061
			243.83	72

Bảng II. Thông tin lưu lượng truy cập nền cho các điểm cuối có tốc độ cao nhất và thấp

Thời lượng điểm cuối		Tổng cộng	nhất Avg. Tốc độ
Kiểu	(tháng) Phiên 2 444.345		phiên (/
Trang chủ	60.979		giây) 5,28
Trường đại học	9		0,19

Bảng III. Lưu lượng tấn công điểm cuối cho hai sâu tốc độ cao và hai tốc độ thấp

Phần mềm độc hại trung bình Tốc độ quét (/giây)	(Các) cổng đã sử dụng
Dloader-NY 46,84	TCP135,139
Forbot-FU 32,53	TCP445
MyDoom-A 0,14	TCP 3127 3198
Robot-AQJ 0,68	TCP139,769

mô phỏng ba loại sâu bổ sung, cụ thể là Witty (sâu có cổng nguồn cố định 4000), CodeRedv2 (sâu có cổng đích cố định 80) và sâu TCP tốc độ thấp (có cổng nguồn cố định và bất thường 2200). Witty và CodeRedv2 được mô phỏng bằng cách sử dụng tốc độ quét, mã giả và tham số được đưa ra trong Shannon và Moore [2004] và Symantec Security.

A.1.3. Bộ dữ liệu giao thông NUST. Lưu lượng truy cập được thu thập tại một bộ định tuyến tại Đại học Khoa học và Công nghệ Quốc gia, Islamabad, Pakistan (NUST), nơi xử lý lưu lượng truy cập từ ba phòng thí nghiệm nghiên cứu riêng biệt. Các phòng thí nghiệm này có tổng cộng khoảng 50 máy chủ. Các cổng được nhân đôi trên bộ định tuyến để nhận lưu lượng (vào, ra và định tuyến nội bộ). Bộ dữ liệu được thu thập vào năm 2009 và lần đầu tiên được sử dụng trong Ali et al. [2010]. Do đó, đây là tập dữ liệu lưu lượng truy cập thực, lớn và gần đây.

Đối với lưu lượng tấn công, quét cổng (TCP SYN), DoS (TCP SYN) và phân mảnh (lũ UDP) đã được khởi chạy đồng thời từ ba máy chủ cuối trong phòng thí nghiệm nghiên cứu. Các cuộc tấn công DoS đã được thực hiện trên hai máy chủ dưới sự quản lý của chúng tôi với các địa chỉ IP công cộng. Mỗi cuộc tấn công được thực hiện trong khoảng thời gian năm phút với địa chỉ IP giả mạo. Đối với mỗi loại tấn công, ba phiên bản tốc độ thấp (0,1, 1, 10 gói/giây) và hai phiên bản tốc độ cao (100, 1000 gói/giây) đã được khởi chạy. Các đặc điểm tấn công cho mỗi cuộc tấn công được thể hiện trong Bảng IV. Lưu lượng truy cập bình thường được ghi lại trong sáu khoảng thời gian, mỗi khoảng thời gian hơn ba gi Trong quá trình nắm bắt lưu lượng, các ứng dụng khác nhau được lưu trữ trên các máy bao gồm truyền tệp, duyệt web, nhắn tin nhanh, truyền phát video theo thời gian thực, v.v.

A.1.4. Bộ dữ liệu ISP. Bộ dữ liệu ISP được thu thập tại Nhà cung cấp dịch vụ Internet (ISP) hàng đầu của Pakistan vào năm 2012. Theo thỏa thuận không tiết lộ, chúng tôi không thể cung cấp bộ dữ liệu công khai. Bộ dữ liệu được thu thập tại bộ định tuyến cổng. Các gói đầy đủ đã được chụp cùng với tải trọng. Nó được thu thập bằng cách phản chiếu cổng tại một công tắc cổng. Công cụ Tcpdump [Công cụ Tcpdump] đã được sử dụng để ghi lưu lượng dành cho cổng 80 và 21 của máy chủ được lưu trữ. Đối với cổng 80, lưu lượng được thu thập trong 3 giờ.

Trung bình 274,37 pkts/giây được quan sát với tốc độ 1,69 mb/giây. Tổng khối lượng lưu lượng được ghi lại là 2,41 GB. Tuy nhiên, không có nhiều hoạt động trên cổng 21 (máy chủ ftp) so với cổng 80, 1GB lưu lượng được thu thập trong 14 giờ.

Bảng IV. Thông tin lưu lượng nền trong các cuộc tấn công

Tên tấn công	Đặc điểm tấn công	Tỷ lệ tấn công (pkts/giây)	Tốc độ lưu lượng nền (pkts/sec)	
			Bộ định tuyến cánh	
				σ
TCP-SYN portcans	Địa chỉ IP đích ngẫu nhiên Đã sửa địa chỉ IP src Hai cuộc tấn công riêng biệt: Lần quét đầu tiên trên cổng 80, Lần quét thứ hai trên cổng 135	0,1	μ 2462,9	474.4
		1	3002.6	398.0
		10	3325.2	397,7
		100	6100.0	2492.4
		1000	3084.7	247.4
TCP-SYN Hai máy chủ từ xa bị tấn công lụt (DoS)	địa chỉ IP src giả mạo Các cổng bị tấn công: 143, 22, 138, 137, 21	0,1	2240.1	216,7
		1	2699.1	328,8
		10	4409.8	1666.2
		100	3964.1	1670.4
		1000	3000,9	238.0
Lũ lụt UDP Hai máy chủ từ xa bị tấn công manh mún	địa chỉ IP src giả mạo Các cổng bị tấn công: 22, 80, 135, 143	0,1	2025.8	506.4
		1	2479.1	291.0
		10	4028.4	1893.1
		100	6565.7	3006.9
		1000	2883.7	260,8

Lưu lượng tấn công được tạo bằng các công cụ như [công cụ Sqlninja], [công cụ Netsparker] và [FTP Brute Force]. Nhiều trường hợp của các cuộc tấn công khác nhau như mật khẩu vũ phu, chèn SQL, leo thang đặc quyền, tải lên tệp tập lệnh, khởi động trình bao đảo ngược, đã được tạo và trộn lẫn trong lưu lượng truy cập sau khi gắn nhãn. Thử nghiệm được thực hiện trong môi trường do ISP cung cấp. Bộ dữ liệu mới, lớn và thực tế, đại diện cho nhiều người dùng của một ISP.

A.2. Bộ dữ liệu dựa trên máy chủ

Đối với các thử nghiệm dựa trên máy chủ, chúng tôi chọn các bộ dữ liệu được sử dụng rộng rãi và có sẵn công khai từ UNM [Bộ dữ liệu UNM] và Phòng thí nghiệm MIT Lincoln (LL) [Bộ dữ liệu MIT]. Chúng tôi cũng sử dụng tập dữ liệu mới hơn từ Nexgin [Bộ dữ liệu Nexgin]. Trước khi chúng tôi cung cấp mô tả về các bộ dữ liệu này, chúng tôi nhấn mạnh các vấn đề với bộ dữ liệu UNM và MIT LL, cũng đã được thảo luận chi tiết trong Merhav et al. [2010]. Cả hai bộ dữ liệu này đều đã khá lỗi thời và ngay cả hệ điều hành được sử dụng để thu thập các bộ dữ liệu này cũng đã thay đổi (tất cả các máy liên quan đều là máy chủ Solaris phiên bản 2.5.1, ngày nay đã cũ). Chúng tôi cũng lưu ý rằng hầu hết các cuộc tấn công trong các bộ dữ liệu này đã lỗi thời và các cuộc tấn công hiện đại phức tạp hơn nhiều. Tuy nhiên, do không có sẵn các bộ dữ liệu được dán nhãn khác, hầu hết các nghiên cứu hiện tại đều sử dụng các bộ dữ liệu này.

A.2.1. Chuỗi cuộc gọi hệ thống UNM. Tập dữ liệu của Đại học New Mexico (UNM) cung cấp dấu vết cuộc gọi hệ thống cho các quy trình khác nhau. Forrest và cộng sự. [1996] lập luận rằng việc giám sát hành vi của một quá trình có thể không bao gồm toàn bộ hành vi bình thường vì một số quá trình hoạt động theo một cách khá đa dạng. Do đó, chúng tạo ra một cách giả tạo các chuỗi cuộc gọi hệ thống. Chúng tôi đã sử dụng dấu vết gửi thư tổng hợp cho các thử nghiệm của mình.

Những dấu vết này được tạo ra bằng cách liệt kê các nguồn biến thể tiềm ẩn đối với các hoạt động gửi thư bình thường. Các tệp theo dõi chứa ID quy trình và các cuộc gọi hệ thống tương ứng của chúng. Đối với dấu vết xâm nhập, chúng tôi sử dụng 3 dấu vết xâm nhập sscp , 2 dấu vết xâm nhập giải mã và 5 dấu vết vòng lặp chuyển tiếp do UNM cung cấp trên trang web của mình. Bảng V cho thấy các cuộc xâm nhập và các phiên bản của chúng được sử dụng; xem [Bộ dữ liệu UNM] để biết chi tiết.

A.2.2. Chuỗi cuộc gọi hệ thống phòng thí nghiệm MIT Lincoln. Từ bộ dữ liệu MIT Lincoln Lab [MIT Dataset], chúng tôi đã sử dụng dữ liệu kiểm toán Solaris BSM, dữ liệu này cung cấp các cuộc gọi hệ thống của Solaris

Bảng V. Các trường hợp xâm nhập UNM

vòng chuyển tiếp giải mã sscp 5		
3	2	

Bảng VI. Trường hợp xâm nhập MIT-LL

httptunnel land ps eject			ftp-write bí mật mailbomb 2			
2		2	2	2	1	2

lưu trữ dưới dạng nhật ký BSM. Tệp BSM chứa thông tin về ID tiến trình, lệnh gọi hệ thống, ID người dùng, tên máy, mô tả và ngày/giờ. Một tệp dữ liệu phân tích lưu lượng mạng riêng biệt cũng được cung cấp cho biết các kết nối mạng gửi đến hệ thống.
Bảng VI liệt kê các trường hợp tấn công trong bộ dữ liệu này; xem [Bộ dữ liệu MIT] để biết chi tiết. Chúng tôi đã lập chỉ mục chéo tệp theo dõi cuộc gọi hệ thống với tệp dữ liệu lưu lượng mạng bằng cách sử dụng các đối số cho lệnh gọi hệ thống exec. Hơn nữa, dung sai một giây đã được chọn theo đề xuất của Kang et al. [2005] để phù hợp với phần lớn các nỗ lực kết nối.
Chúng tôi đã sử dụng dữ liệu lành tính từ tuần đầu tiên (01/03/1999) và các cuộc tấn công được dán nhãn từ tuần thứ hai (08/03/1999).

A.2.3. Bộ dữ liệu Nexgin. Đối với các ADS hoạt động trên chuỗi lệnh gọi Windows API, chúng tôi đã sử dụng bộ dữ liệu Nexgin [Bộ dữ liệu Nexgin]. Bộ dữ liệu Nexgin được thu thập bằng cách cắt exe các tệp thực thi lành tính và độc hại trên máy Microsoft Windows XP và trình theo dõi lệnh gọi API thương mại được sử dụng để ghi nhật ký các lệnh gọi API được thực hiện bởi các quy trình sử dụng hook chế độ nhân. Để giảm bớt sự phức tạp của việc theo dõi thời gian chạy, các tác giả đã liệt kê ngắn gọn 237 lệnh gọi API lõi từ sáu danh mục chức năng khác nhau như ổ cắm, quản lý bộ nhớ, quy trình và luồng, v.v. Các tác giả đã thu thập nhật ký lệnh gọi hệ thống của 100 chương trình lành tính, 117 trojan, 165 virus và 134 sâu. Các mẫu phần mềm độc hại được cung cấp trong bộ dữ liệu này sử dụng danh pháp thích hợp cho phần mềm độc hại, giúp dễ dàng xác định họ và các biến thể của phần mềm độc hại khác nhau.

B. HỆ THỐNG PHÁT HIỆN SỰ THẬT

Trong phụ lục này, chúng tôi mô tả ngắn gọn các thuật toán phát hiện bất thường được sử dụng để đánh giá độ chính xác. Trước khi mô tả các ADS thời gian thực được sử dụng trong công việc này, chúng tôi xin nhắc lại rằng thuật toán thích ứng ngưỡng thực tế không nên dành riêng cho một ADS cụ thể. Do đó, mặc dù chúng tôi đã chọn một số ADS để đo điểm chuẩn bằng chứng về khái niệm và hiệu suất, nhưng tất cả các phân tích và đặc điểm sẽ được cung cấp trong các phần tiếp theo là chung chung và phải có trên các ADS thời gian thực khác nhau. Chúng tôi cũng nhấn mạnh rằng các ADS được sử dụng trong công việc này khá đa dạng về các nguyên tắc và tính năng phát hiện cơ bản của chúng.

Tất cả các ADS được sử dụng trong công việc này đã được đào tạo bằng 50% tổng số dữ liệu; ing 50% còn lại được sử dụng để thử nghiệm. Để đo điểm chuẩn hiệu suất, chúng tôi thay đổi ngưỡng phân loại bất thường của tất cả các trình phát hiện để tạo các đường cong ROC. Tất cả các tham số khác của ADS giống như được báo cáo trong Forrest et al. [1996], Gu et al. [2005], Jung và cộng sự. [2004], Wang và Stolfo [2004], Kang và cộng sự. [2005], Mahoney và Chan [2001], Sung et al. [2004], và Gao et al. [2005].

Để sử dụng ngưỡng thích ứng với các ADS, chuỗi markov bậc 15 được sử dụng vì hầu hết các ADS hiển thị entropy có điều kiện thấp theo thứ tự này. Mặc dù một số máy dò có thể hoạt động theo thứ tự thấp hơn nhưng để đơn giản, chúng tôi đã sử dụng cùng một thứ tự. Bộ lọc trung vị đã được áp dụng trên 15 giá trị để loại bỏ các thay đổi lẻ tẻ. Do đó, ma trận xác suất chuyển đổi có thứ tự 15 × 15. Đối với mỗi ADS, điểm bất thường tối thiểu và tối đa được tính trên dữ liệu huấn luyện. Những điểm bất thường này được ánh xạ tới 15 thùng có kích thước bằng nhau. Thùng đầu tiên và thùng cuối cùng được giữ linh hoạt để phù hợp với điểm số bất thường chưa từng thấy, nghĩa là lớn hơn mức tối đa hoặc nhỏ hơn mức tối thiểu.

Do đó, rn trong biểu thức. (3) là thùng đại diện cho điểm bất thường. Các tham số khác của thuật toán, phương trình. (4) và (5), được tính toán dựa trên thùng được quan sát. Tuy nhiên, ma trận xác suất chuyển đổi được cập nhật và dự đoán thùng tiếp theo bằng cách sử dụng các phương trình. (6) và (7). Bây giờ chúng tôi thảo luận ngắn gọn về các ADS được sử dụng trong công việc này; độc giả được tham khảo các bài báo gốc để biết chi tiết về các thuật toán này.

B.1. Quảng cáo mạng

Trong phần này, chúng tôi thảo luận ngắn gọn về các hệ thống phát hiện bất thường dựa trên mạng.

Máy dò dị thường Entropy cực đại [Gu et al. 2005]. Nó ước tính phân phối lưu lượng lành tính bằng cách sử dụng ước tính entropy tối đa. Lưu lượng được chia thành 2348 lớp gói. Phân phối theo kinh nghiệm đã được tính toán cho lưu lượng đào tạo. Bộ công cụ TADM [TADM toolkit] sau đó đã được sử dụng để ước lượng tham số cho mô hình entropy cực đại.
Ước tính tham số này cùng với phân phối đã được sử dụng để tạo phân phối cơ sở. Các phân phối lớp gói được quan sát trong các cửa sổ thời gian thực sau đó được so sánh với phân phối cơ sở bằng thước đo phân kỳ Kullback-Leibler (KL).
Nếu phân phối xác suất của một lớp cụ thể trong phân phối cơ sở là 0, nó sẽ dẫn đến phân kỳ KL vô hạn. Để tránh điều này, một giá trị lớn (lớn hơn ngưỡng) đã được sử dụng cho phân kỳ KL. Một cảnh báo đã được đưa ra nếu phân kỳ KL của lớp gói vượt quá ngưỡng, k, hơn h lần trong W cửa sổ cuối cùng của t giây mỗi lần. ROC được tạo bằng cách thay đổi k. Tuy nhiên, số lượng cửa sổ W được đặt thành 60 và mỗi cửa sổ t có kích thước 1 giây. Giá trị của h được đặt thành 30. Để đạt được điều này, ngưỡng thích ứng được áp dụng trên phân kỳ KL của lớp gói để dự đoán ngưỡng cứ sau t giây dựa trên điểm phân kỳ KL được quan sát trước đó cho lớp đó.

Thuật toán bước đi ngẫu nhiên ngưỡng (TRW) [Jung et al. 2004]. TRW phát hiện các lần quét cổng đến bằng cách lưu ý rằng xác suất nỗ lực kết nối thành công đối với một máy chủ lành tính sẽ cao hơn nhiều so với đối với một máy quét. Để thúc đẩy quan sát này, TRW sử dụng kiểm tra giả thuyết tuần tự (nghĩa là kiểm tra tỷ lệ khả năng để phân loại xem máy chủ từ xa có phải là máy quét hay không). Tỷ lệ khả năng này được so sánh với hai ngưỡng; giới hạn trên và dưới, η1 và η0 tương ứng. Nếu tỷ lệ khả năng lớn hơn η1, máy chủ từ xa được phân loại là máy quét và nếu nó nhỏ hơn η0, nó được phân loại là lành tính. Tuy nhiên, nếu tỷ lệ khả năng không cao hơn η1 và cũng không thấp hơn η0, thì nó sẽ đợi nhiều mẫu hơn cho đến khi tỷ lệ khả năng đạt đến một trong các giới hạn này. Nó đặt cho PD các giới hạn trên các giới hạn, chẳng hạn như η1 ≤ trong đó PD và PF lần lượt là xác suất dương của PF phát hiện và sai . Nó báo cáo rằng nó không phải là một tiên nghiệm rõ ràng làm thế nào để chọn các ngưỡng này. Tuy nhiên, xác suất PD và PF có thể được thay thế bằng người dùng cho sen β và α tương ứng. Cuối cùng, ROC được tạo bằng cách thay đổi giá trị của các giới hạn này. Tuy nhiên, ngưỡng thích ứng đã được áp dụng ở giới hạn trên η1 của tỷ lệ khả năng, nghĩa là nếu tỷ lệ thực tế vượt quá tỷ lệ dự đoán (được đặt làm giới hạn trên), thì máy chủ sẽ được phân loại là máy quét. Giới hạn dưới η0 được giữ cố định. Ngưỡng thích ứng được sử dụng để dự đoán tỷ lệ khả năng xảy ra cho mọi trường hợp dựa trên các trường hợp trước, do đó được đặt làm giới hạn trên.

Phát hiện bất thường tiêu đề gói (PHAD) [Mahoney và Chan 2001]. PHAD tìm hiểu phạm vi giá trị thông thường cho tất cả 33 trường trong tiêu đề Ethernet, IP, TCP, UDP và ICMP. Điểm bất thường được chỉ định cho từng trường tiêu đề gói trong giai đoạn thử nghiệm và điểm của các trường được tính tổng để có được điểm bất thường tổng hợp của gói. Chúng tôi đánh giá PHAD-C32 [Mahoney và Chan 2001] bằng cách sử dụng các trường tiêu đề gói sau: IP nguồn, IP đích, cổng nguồn, cổng đích, loại giao thức và cờ TCP. Các giá trị n hàng đầu được ngưỡng là bất thường. ROC được tạo bằng cách thay đổi giá trị của n, tuy nhiên, ngưỡng thích ứng được đặt trên điểm bất thường tổng hợp của gói. Do đó, giá trị của n không cố định và được thực hiện thích ứng.

Phát hiện bất thường dựa trên tải trọng (PAYL) [Wang và Stolfo 2004]. PAYL tính toán phân phối tần số byte và độ lệch chuẩn của tải trọng ứng dụng truyền đến máy chủ và cổng bằng mô hình n-gram. Ở đây n được đặt thành 1 vì nó được thực hiện cho số lần xuất hiện của mỗi byte trong tải trọng. Điều này được thực hiện riêng cho tất cả các độ dài tải trọng được quan sát. Để học cấu hình bình thường một cách nhanh chóng, học gia tăng được sử dụng để cập nhật phân phối. Vì nó được thực hiện cho tất cả các độ dài tải trọng được quan sát, nên nó mang lại một số lượng lớn các mô hình. Phân cụm được thực hiện để giảm số lượng mô hình bằng cách so sánh các mô hình lân cận (các thùng có chiều dài i - 1 và i + 1) sử dụng khoảng cách Manhattan. Nếu khoảng cách nhỏ hơn ngưỡng t (được đặt thành 0,5), thì các mô hình được hợp nhất bằng cách sử dụng phương pháp học tăng dần. Vì lưu lượng không được làm sạch và có thể chứa các cuộc tấn công nên phương pháp học không giám sát đã được triển khai để loại bỏ nhiễu. Mô hình đã học được áp dụng cho dữ liệu huấn luyện để xác định các điểm ngoại lệ và quá trình huấn luyện được thực hiện lại sau khi loại bỏ các điểm ngoại lệ. Các ngoại lệ được xác định bằng cách sử dụng nguyên tắc phát hiện của thuật toán, đó là tính toán khoảng cách mahalanobis giữa mô hình được đào tạo và gói được quan sát. Nếu khoảng cách lớn hơn ngưỡng d (được đặt thành 256), các gói được phân loại là ngoại lệ. Trong giai đoạn phát hiện, có khả năng độ dài quan sát không được quan sát trong quá trình đào tạo. Trong trường hợp đó, phân phối thời gian chạy được so sánh với mô hình lân cận có độ dài gần nhất. Để tránh khoảng cách vô hạn trong trường hợp độ lệch chuẩn bằng 0 (điều này có thể xảy ra nếu một byte không bao giờ xuất hiện trong quá trình huấn luyện hoặc xuất hiện với tần suất chính xác như nhau trong mỗi mẫu), hệ số làm mịn 0,001 đã được thêm vào độ lệch chuẩn. Để tạo ROC, ngưỡng khoảng cách mahalanobis d đã được thay đổi để quan sát tỷ lệ phát hiện/dương tính giả khác nhau. Tuy nhiên, ngưỡng thích ứng đã được áp dụng trên khoảng cách ma halanobis và d được đặt một cách thích ứng dựa trên các mẫu trước đó được quan sát có cùng độ dài.

B.2. ADS lưu trữ

Trong phần này, chúng tôi thảo luận ngắn gọn về các hệ thống phát hiện bất thường dựa trên máy chủ.

Sequence Time Delay Embedding (STIDE) [Forrest et al. 1996]. STIDE phát hiện các cuộc gọi hệ thống bất thường của một quá trình xâm nhập. Đầu tiên, STIDE xây dựng một cấu hình bình thường cho một quy trình bằng cách trượt một cửa sổ có kích thước k + 1 qua dấu vết của các cuộc gọi hệ thống và ghi lại cuộc gọi nào theo sau cuộc gọi nào trong cửa sổ trượt. Khi chúng tôi có cơ sở dữ liệu về các mẫu, chúng tôi kiểm tra dấu vết mới đối với nó bằng cách trượt dấu vết mới với kích thước cửa sổ k + 1 đối với cơ sở dữ liệu. Nếu chuỗi các cuộc gọi hệ thống khác với chuỗi được ghi trong cơ sở dữ liệu thông thường, thì dấu vết sẽ tạo ra sự không khớp. Chúng tôi ghi lại số lần không khớp theo tỷ lệ phần trăm của tổng số lần không khớp có thể xảy ra. Đối với các thử nghiệm của mình, chúng tôi đã sử dụng k = 6. Giá trị ngưỡng được áp dụng cho phần trăm không khớp để phân loại hành vi bất thường. Sau đó, thuật toán ngưỡng thích ứng được áp dụng để dự đoán điểm bất thường (tỷ lệ phần trăm không khớp) cho chuỗi lệnh gọi hệ thống tiếp theo. Ví dụ: đầu ra từ mỗi lần theo dõi xâm nhập lprcp được sử dụng để dự đoán đầu ra cho lần theo dõi xâm nhập lprcp tiếp theo.

Support Vector Machines (SVM) Sử dụng phương pháp gọi túi hệ thống [Kang et al. 2005]. Kang và cộng sự. [2005] đề xuất một túi biểu diễn cuộc gọi hệ thống để phát hiện chuỗi cuộc gọi hệ thống xâm nhập. Trong quá trình chuyển đổi sang gói gửi lại cuộc gọi hệ thống, tần suất của mỗi cuộc gọi hệ thống trong túi được giữ nguyên và thông tin đặt hàng giữa các cuộc gọi hệ thống sẽ bị mất. Mỗi phiên bản thực thi quy trình thông thường được sử dụng để chỉ tạo một tính năng được xác định là danh sách có thứ tự về tần suất của tất cả các lệnh gọi hệ thống trong chuỗi thực thi đó. Sau đó, phiên bản không xác định được so sánh với các phiên bản bình thường để tạo điểm phân loại bằng SVM. Tương tự như STIDE, thuật toán ngưỡng thích ứng sau đó sử dụng đầu ra từ một dấu vết xâm nhập để dự đoán điểm bất thường (điểm phân loại SVM) cho dấu vết xâm nhập tiếp theo.

Khoảng cách hành vi (Bdist) [Gao et al. 2005]. Cách tiếp cận Khoảng cách hành vi hoạt động dựa trên giả thuyết rằng phải có sự giống nhau về ngữ nghĩa giữa hai bản sao của một quy trình khi đầu vào của các quy trình giống nhau. Để xác định sự thay đổi hành vi giữa các bản sao của một quy trình, các tác giả đã liên kết một cụm từ cuộc gọi hệ thống cụ thể (chuỗi các cuộc gọi hệ thống thường xuất hiện cùng nhau trong quá trình thực thi chương trình) do một bản sao phát ra với một cụm từ cuộc gọi hệ thống do bản sao kia phát ra. Mối tương quan được tính bằng cách sử dụng thước đo khoảng cách gọi là khoảng cách tiến hóa, là tổng chi phí thay thế, xóa, chèn và lệch. Số đo khoảng cách được tính toán bằng cách sử dụng các cách sắp xếp khác nhau và cách căn chỉnh dẫn đến khoảng cách hành vi nhỏ nhất đã được chọn. Để trích xuất và rút gọn cụm từ, các thuật toán tương tự đã được sử dụng bởi Gao et al. [2005] (tức là TEIRESIAS) đã được sử dụng. Đối với các thử nghiệm của mình, chúng tôi đã sử dụng bộ dữ liệu Nexgin để xác định các biến thể của phần mềm độc hại. Cuối cùng, ngưỡng đã được áp dụng cho điểm số khoảng cách hành vi của mọi cụm từ lệnh gọi API. Xin lưu ý rằng mặc dù Gao et al. [2005] chỉ cung cấp kết quả đánh giá trên các lệnh gọi hệ thống Linux, họ đã tuyên bố rõ ràng trong bài viết của mình rằng phương pháp này đã được đề xuất cho cả các lệnh gọi hệ thống Linux và lệnh gọi Windows API và chúng tôi sử dụng nó trên các lệnh gọi Windows API. Đối với ngưỡng thích ứng, đầu ra của một biến thể phần mềm độc hại đã được sử dụng để dự đoán giá trị ngưỡng điểm bất thường tiếp theo (ngưỡng điểm khoảng cách). Ví dụ: đầu ra ngưỡng từ w32.Mydoom.A đã được sử dụng để dự đoán ngưỡng cho một phiên bản đa hình khác của W32.Mydoom.A.

Trình phân tích tĩnh cho các tệp thực thi xấu (SAVE) [Sung et al. 2004]. SAVE phát hiện phần mềm độc hại hợp nhất (hoặc đa hình) và đột biến (hoặc biến chất). SAVE hoạt động dựa trên tiền đề đơn giản là tất cả các phiên bản của cùng một phần mềm độc hại đều có chung một chữ ký cốt lõi là sự kết hợp của một chuỗi lệnh gọi API của phần mềm độc hại. Chuỗi lệnh gọi API được trích xuất từ mẫu phần mềm độc hại. Trình tự này cung cấp cơ sở để phát hiện các biến thể và đột biến của cùng một phần mềm độc hại. Để so sánh một trình tự không xác định với trình tự API đã trích xuất, các trình tự này trước tiên được căn chỉnh và sau đó là phép đo Cosine, phép đo Jaccard mở rộng và các phép đo tương quan Pearson được sử dụng chung để xác định xem mẫu không xác định có độc hại hay không. Đối với các thử nghiệm của chúng tôi, chúng tôi lấy giá trị trung bình của giá trị ngưỡng của phép đo Cosine, giá trị ngưỡng của phép đo Jackard mở rộng và giá trị ngưỡng của phép đo tương quan Pearson (như trong bài báo gốc). Để sử dụng thuật toán ngưỡng thích ứng, tương tự như Bdist, đầu ra ngưỡng từ một mẫu đa hình được sử dụng để dự đoán giá trị ngưỡng cho mẫu tiếp theo.

NGƯỜI GIỚI THIỆU

Aggarwal, CC, Han, J., Wang, J., và Yu, PS 2006. Khung phân loại theo yêu cầu về tiến hóa các luồng dữ liệu. IEEE Trans. hiểu biết. Dữ liệu Eng. 18, 5, 577–589.

Agosta, JM, Wasser, CD, Chandrashekar, J., và Livadas, C. 2007. Máy dò dị thường thích nghi để phát hiện sâu. Trong Kỷ yếu Hội thảo USENIX lần thứ 2 về Xử lý các Sự cố Hệ thống Máy tính bằng Kỹ thuật Máy học. Hiệp hội USENIX, Berkeley, CA, 3:1–3:6.

Ali, MQ, Khan, H., Sajjad, A., và Khayam, SA 2009. Về việc đạt được điểm vận hành tốt trên mặt phẳng ROC bằng cách sử dụng dự đoán điểm bất thường ngẫu nhiên. Trong Kỷ yếu của Hội nghị ACM lần thứ 16 về Bảo mật Máy tính và Truyền thông (CCS'09). ACM, New York, 314–323.

Ali, S., Haq, I., Rizvi, S., Rasheed, N., Sarfraz, U., Khayam, SA, và Mirza, F. 2010. Về giảm thiểu việc mất độ chính xác do lấy mẫu trong các hệ thống phát hiện bất thường giao thông. Máy tính ACM SIGCOMM. cộng đồng. Rev. 40, 4–16.

Arbor PeakFlow. Sản phẩm lưu lượng đỉnh của mạng Arbor. http://www.arbornetworks.com/peakflowsp.

Ashfaq, AB, Joseph, M., Mumtaz, A., Ali, MQ, Sajjad, A., và Khayam, SA 2008. Đánh giá so sánh các thiết bị phát hiện dị thường dưới các cuộc tấn công quét cổng. Trong Kỷ yếu của Hội nghị chuyên đề quốc tế lần thứ 11 về những tiến bộ gần đây trong phát hiện xâm nhập (RAID'08). Springer-Verlag, Berlin, 351–371.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., và Gavalda, R. 2009. Các phương pháp tập hợp mới để phát triển các luồng dữ liệu. Trong Kỷ yếu của Hội nghị Quốc tế ACM SIGKDD lần thứ 15 về Khám phá Tri thức và Khai thác Dữ liệu (KDD'09). ACM, 139–148.

Cardenas, AA, Baras, JS, và Seamon, K. 2006. Khung đánh giá các hệ thống phát hiện xâm nhập. Trong Kỷ yếu của Hội nghị chuyên đề IEEE về Bảo mật và Quyền riêng tư (SP'06). IEEE.

Chen, S., Wang, H., Zhou, S. và Yu, PS 2008. Ngừng chạy theo xu hướng: Khám phá các mô hình bậc cao trong dữ liệu đang phát triển. Trong Kỷ yếu của Hội nghị Quốc tế về Kỹ thuật Dữ liệu lần thứ 24 của IEEE năm 2008 (ICDE'08). Hiệp hội máy tính IEEE, Los Alamitos, CA, 923–932.

Bảo vệ bất thường của Cisco. Trang chủ mô-đun bảo vệ bất thường của Cisco. www.cisco.com/en/US/products/ps6235/.

Cretu-Ciocarlie, GF, Stavrou, A., Locasto, ME, và Stolfo, SJ 2009. Phát hiện sự bất thường thích ứng thông qua tự hiệu chỉnh và cập nhật động. Trong Kỷ yếu của Hội nghị chuyên đề quốc tế lần thứ 12 về những tiến bộ gần đây trong phát hiện xâm nhập (RAID'09). Springer-Verlag, Berlin, 41–60.

Crovella, ME, và Bestavros, A. 1997. Tính tự tương tự trong lưu lượng truy cập web trên toàn thế giới: Bằng chứng và nguyên nhân có thể. Bộ chuyển đổi IEEE/ACM. mạng. 5, 835–846.

Forrest, S., Hofmeyr, SA, Somayaji, A., và Longstaff, TA 1996. Ý thức về bản thân đối với các quy trình unix. Trong Kỷ yếu của Hội nghị chuyên đề về Bảo mật và Quyền riêng tư của IEEE năm 1996 (SP'96). Hiệp hội máy tính IEEE, Los Alamitos, CA, 120–128.

FTP Brute Force. Ssh2ftpcrack ftp/ssh vũ phu. http://packetstormsecurity.org/files/98155/SSH2FTPCrack-FTP-SSH-Brute-Forcer.html.

Gao, D., Reiter, MK, và Song, D. 2005. Khoảng cách hành vi để phát hiện xâm nhập. Trong Kỷ yếu của Hội nghị chuyên đề quốc tế lần thứ 8 về những tiến bộ gần đây trong phát hiện xâm nhập (RAID). 63–81.

Gao, J., Fan, W. và Han, J. 2007. Về các giả định phù hợp để khai thác các luồng dữ liệu: Phân tích và thực hành. Trong Kỷ yếu của Hội nghị Quốc tế IEEE lần thứ 7 về Khai thác Dữ liệu (ICDM'07). Hiệp hội máy tính IEEE, Los Alamitos, CA, 143–152.

Báo cáo Gartner. 2003. Chu kỳ thổi phồng về bảo mật thông tin của Gartner tuyên bố các hệ thống phát hiện xâm nhập đã ngừng hoạt động Nếu thất bại, số tiền dành cho việc phát hiện xâm nhập nên được đầu tư vào tường lửa. http://www.gartner.com/about/press release/pr11june2003c.jsp.

Gu, G., Fogla, P., Dagon, D., Lee, W., và Skoric, B. 2006. Hướng tới khung lý thuyết thông tin để phân tích các hệ thống phát hiện xâm nhập. Trong Kỷ yếu của Hội nghị chuyên đề Châu Âu lần thứ 11 về Nghiên cứu An ninh Máy tính (ESORICS'06).

Gu, Y., McCullum, A., và Towsley, D. 2005. Phát hiện sự bất thường trong lưu lượng mạng bằng ước tính entropy tối đa. Trong Kỷ yếu của Hội nghị ACM SIGCOMM lần thứ 5 về Đo lường Internet (IMC'05). Hiệp hội USENIX, Berkeley, CA, 32–32.

Hồ sơ đống. HPROF: Công cụ định hình heap/CPU trong j2se5.0. http://docs.oracle.com/javase/7/docs/technotes/samples/hprof.html.

Hollinger, G., Djugash, J. và Singh, S. 2008. Theo dõi mục tiêu đang di chuyển trong môi trường lộn xộn với các đài khác nhau: Kết quả mở rộng. Công nghệ. trả lời. CMU-RI-TR-08-07, Viện Robotics, Đại học Carnegie Mellon.

Ide, T. và Kashima, H. 2004. Phát hiện bất thường dựa trên không gian riêng trong các hệ thống máy tính. Trong Kỷ yếu của Hội nghị Quốc tế ACM SIGKDD lần thứ 10 về Khám phá Tri thức và Khai thác Dữ liệu (KDD'04). ACM, New York, 440–449.

Jung, J., Paxson, V., Berger, AW, và Balakrishnan, H. 2004. Phát hiện quét cổng nhanh bằng cách sử dụng thử nghiệm giả thuyết tuần tự. Trong Kỷ yếu của Hội nghị chuyên đề IEEE về Bảo mật và Quyền riêng tư (SP'04). Hiệp hội máy tính IEEE, Los Alamitos, CA.

Kang, DK, Fuller, D., và Honavar, V. 2005. Học phân loại để phát hiện sai mục đích và bất thường bằng cách sử dụng một túi biểu diễn cuộc gọi hệ thống. Trong Kỷ yếu của Hội thảo Đảm bảo Thông tin Điều khiển và Con người của Hệ thống IEEE lần thứ 6 (IAW'05).

Kolter, JZ, và Maloof, MA 2005. Sử dụng nhóm chuyên gia phụ gia để đối phó với sự trôi dạt khái niệm. Trong Kỷ yếu của Hội nghị Quốc tế lần thứ 22 về Học máy (ICML'05). ACM, New York, 449–456.

Lakhina, A., Crovella, M., và Diot, C. 2004. Chẩn đoán sự bất thường về lưu lượng trên toàn mạng. Trong Kỷ yếu của Hội nghị về Ứng dụng, Công nghệ, Kiến trúc và Giao thức cho Truyền thông Máy tính (SIGCOMM'04). ACM, New York, 219–230.

Lakhina, A., Crovella, M., và Diot, C. 2005. Khai thác dị thường bằng cách sử dụng phân phối tính năng lưu lượng truy cập. Trong Kỷ yếu của Hội nghị về Ứng dụng, Công nghệ, Kiến trúc và Giao thức cho Truyền thông Máy tính (SIGCOMM'05). ACM, New York, 217–228.

Bộ dữ liệu LBNL. Dự án truy vết doanh nghiệp LBNL/ICSI. http://www.icir.org/enterprise-tracing/Overview.html.

Lippmann, RP, Haines, JW, Fried, DJ, Korba, J. và Das, K. 2000. Đánh giá phát hiện xâm nhập ngoại tuyến DARPA năm 1999. Điện toán. mạng. 34, 579–595.

Mahoney, MV and Chan, PK 2001. PHAD: Phát hiện bất thường tiêu đề gói để xác định thù địch lưu lượng mạng. Công nghệ. trả lời. CS-2001-4, Công nghệ Florida.

Masud, MM, Chen, Q., Khan, L., Aggarwal, C., Gao, J., Han, J., và Thuisingham, B. 2010. Đề cập đến sự tiến hóa của khái niệm trong các luồng dữ liệu trôi dạt về khái niệm. Trong Kỷ yếu của Hội nghị Quốc tế IEEE về Khai thác Dữ liệu (ICDM'10) IEEE Computer Society, Los Alamitos, CA, 929–934.

Masud, MM, Gao, J., Khan, L., Han, J., và Thu Raisingham, BM 2011. Phân loại và phát hiện lớp mới trong luồng dữ liệu trôi dạt khái niệm dưới các ràng buộc về thời gian. IEEE Trans. hiểu biết. Dữ liệu Eng. 23, 6, 859–874.

Merhav, M., Gutman, M., và Ziv, J. 1989. Về ước tính thứ tự của chuỗi markov và phổ quát nén dữ liệu. IEEE Trans. thông tin liên lạc Lý thuyết 35, 5, 1014–1019.

Merhav, M., Gutman, M., và Ziv, J. 2010. Phát hiện xâm nhập thông qua trình tự cuộc gọi hệ thống và đối số Phân tích. IEEE Trans. Phụ thuộc. Máy tính an toàn. 7, 4.

Bộ dữ liệu MIT. Phòng thí nghiệm MIT Lincoln, công nghệ hệ thống thông tin. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html.

Moore, D., Shannon, C., và Claffy, K. 2002. Code-red: Nghiên cứu tình huống về sự lây lan và nạn nhân của sâu máy tính. Trong Code-Red: Nghiên cứu điển hình về sự lây lan và nạn nhân của sâu Internet, ACM, New York.

Công cụ Netsparker. Netsparker, trình quét bảo mật ứng dụng web. http://www.mavitunasecurity.com/netsparker/.

Bộ dữ liệu Nexgin. Bộ dữ liệu Nexgin RC. http://www.nexginrc.org/Datasets/Default.aspx.

Bộ dữ liệu NUST. Bộ dữ liệu giao thông NUST. http://wisnet.seecs.nust.edu.pk/projects/nes/datasets.html.

Pang, R., Allman, M., Bennett, M., Lee, J., Paxson, V., và Tierney, B. 2005. Cái nhìn đầu tiên về lưu lượng giải thưởng đầu vào hiện đại. Trong Kỷ yếu của Hội nghị ACM SIGCOMM lần thứ 5 về Đo lường Internet (IMC'05). Hiệp hội USENIX, Berkeley, CA, 2–2.

Ryu, YU và Rhee, HS 2008. Đánh giá các hệ thống phát hiện xâm nhập dưới một ràng buộc tài nguyên. ACM Dịch. thông tin liên lạc hệ thống. bảo mật. 11, 20:1–20:24.

Shannon, C. và Moore, D. 2004. Sự lây lan của loài sâu hóm hỉnh. Trong Kỷ yếu về Bảo mật và Quyền riêng tư của IEEE (SP'04) 2, 46–50.

Công cụ Sqlninja. Sqlninja, một công cụ tiếp quản và xâm nhập máy chủ SQL. http://sqlninja.sourceforge.net/.

Sung, AH, Xu, J., Chavez, P., và Mukkamala, S. 2004. Bộ phân tích tĩnh các tệp thực thi xấu (SAVE). Trong Kỷ yếu của Hội nghị Ứng dụng Bảo mật Máy tính Thường niên lần thứ 20 (ACSAC'04). Hiệp hội máy tính IEEE, Los Alamitos, CA, 326–334.

Bảo mật Symantec. Phản ứng bảo mật của Symantec. http://securityresponse.symantec.com/avcenter.

Bộ công cụ TADM. Tadm, bộ công cụ cho mô hình phân biệt nâng cao. http://tadm.sourceforge.net.

Công cụ tcpdump. Kho lưu trữ công cộng Tcpdump/libpcap. http://www.tcpdump.org/.

Trees, HLV 2001. Lý thuyết Phát hiện, Ước tính và Điều chế: Phần I 1st Ed. Wiley-Liên khoa học.

Twycross, J. và Williamson, MM 2003. Triển khai và thử nghiệm bộ điều tiết vi-rút. Trong Kỷ yếu của Hội nghị lần thứ 12 về USENIX Security Symposium. Hiệp hội USENIX, Berkeley, CA, 20–20.

Bộ dữ liệu UNM. Hệ thống miễn dịch máy tính, bộ dữ liệu. http://www.cs.unm.edu/ immsec/data/synth-sm.html.

Wang, K. và Stolfo, SJ 2004. Phát hiện xâm nhập mạng dựa trên tải trọng bất thường. Trong Kỷ yếu của Hội nghị chuyên đề quốc tế lần thứ 7 về những tiến bộ gần đây trong phát hiện xâm nhập (RAID). 203–222.

QUẢNG CÁO WisNet. Trang chủ so sánh Wisnet ADS. http://wisnet.niit.edu.pk/projects/adeval.

Yu, Z., Tsai, JJP, và Weigert, T. 2007. Hệ thống phát hiện xâm nhập tự động điều chỉnh. IEEE Trans. Hệ thống, Con người và Cybernet. 37, 373–384.

Yu, Z., Tsai, JJP, và Weigert, T. 2008. Hệ thống phát hiện xâm nhập tự động điều chỉnh thích ứng. ACM Trans. tự chủ. Hệ thống thích ứng 3, 10:1–10:25.

Nhận tháng 2 năm 2011; sửa đổi tháng 9 năm 2011, tháng 3 năm 2012, tháng 10 năm 2012; chấp nhận tháng 1 năm 2013