TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 03(40) (2020) 16-25

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Tự động hóa việc cập nhật bảng luật lọc gói cho các firewall mã nguồn mở

Automating the update of packet filtering rules for open source firewalls

Nguyễn Kim Tuấn[a,b*], Nguyễn Trung Thuận[a,b], Phan Long[a,b]
Kim Tuan Nguyen[a,b*]; Trung Thuan Nguyen[a,b]; Long Phan[a,b]

[a]*Khoa Công nghệ Thông tin, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*
[b]*Viện Nghiên cứu và Phát triển Công nghệ Cao, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*
[a]*Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Vietnam*
[b]*Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam*

(Ngày nhận bài: 20/05/2020, ngày phản biện xong: 12/06/2020, ngày chấp nhận đăng: 27/6/2020)

Tóm tắt

Trong các hệ thống bảo mật mạng doanh nghiệp dựa vào Intrusion Detection System (IDS) và Firewall (tường lửa bảo mật), thông thường, nhiệm vụ của IDS là theo dõi tất cả dòng traffic vào/ra mạng, để từ đó phát hiện và đưa ra cảnh báo về các dòng traffic bất thường, về các gói tin có nguy cơ mang theo mã độc. Dựa vào những những thông tin cảnh báo này, người quản trị an ninh mạng sẽ thiết kế các luật chính sách mới, tiến hành cập nhật lại bảng luật lọc gói của Firewall, nhờ đó firewall có thể ngăn chặn kịp thời các dòng traffic không mong muốn. Trong bài báo này, chúng tôi đề xuất và triển khai một mô hình mới về sự kết hợp trong hoạt động giữa IDS và Firewall, nó cho phép IDS tự động cập nhật bảng luật lọc gói của firewall mỗi khi phát hiện có sự bất thường. Điều này giúp việc bảo vệ mạng trước các tấn công từ Internet trở nên tức thời hơn.

Từ khóa: Tường lửa bảo mật; Bảng luật lọc gói; Hệ thống phát hiện xâm nhập; Nghe lén; Lọc gói tin.

Abstract

In enterprise network security systems that rely on IDS (Intrusion Detection System) and Firewall, it is the typical task of IDS to monitor all incoming/outgoing network traffic, thereby detecting and giving warnings about abnormal traffic streams, packets which are at risk of bringing malicious code. Based on the warning information, the network system administrator will design new policy rules, update the packet filtering rules of the Firewall, so that the firewall can prevent unwanted traffic in time. In this paper, we propose and deploy a new model of the association between the operation of IDS and Firewall, which allows IDS to automatically update the firewall filter rule table whenever it detects anomalies. This helps to protect the network from Internet attacks.

Keywords: Firewall; rule table; intrusion detection system; sniff; packet capture.

1. Đặt vấn đề

Trong mô hình bảo mật mạng doanh nghiệp, hệ thống phát hiện xâm nhập (IDS) đóng vai trò quan trọng. Nó có nhiệm vụ giám sát mọi dòng traffic, mọi packet vào/ra mạng, để từ đó có thể phát hiện ra các xâm nhập trái phép và các packet, có thể đến từ các nguồn hợp pháp, có nguy cơ mang theo mã độc vào mạng [1].

**Corresponding Author: Kim Tuan Nguyen; Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Vietnam; Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam.*
Email: nguyenkimtuan@duytan.edu.vn

Khi đã xác định được hành động bất thường từ một traffic nào đó thì IDS tiến hành ngay các công việc cần thiết như lưu giữ thông tin liên quan đến sự bất thường, hay gửi thông tin cảnh báo về sự bất thường đến các hệ thống liên quan, trong đó có cả người quản trị an ninh của mạng. Cần lưu ý rằng, IDS chỉ làm nhiệm vụ phát hiện và đưa ra cảnh báo, còn việc xử lý sự bất thường là do các bộ phận khác. Điều này thường được thực hiện bởi firewall thông qua sự điều khiển bảng luật của người quản trị an ninh mạng. Rõ ràng công đoạn này có vẻ thụ động, cần có thời gian để cập nhật bảng luật, nên cần được xem xét cải tiến để nâng cao hiệu quả trong công tác bảo vệ mạng của tổ chức, doanh nghiệp [2].

Có thể nói, bộ lọc gói và bảng luật lọc gói là hai trong ba thành phần chính của các firewall lọc gói. Từ chính sách điều khiển truy cập mạng, đội quản trị an ninh mạng sẽ xây dựng bảng luật lọc gói cho firewall. Bộ lọc gói dựa vào bảng luật này để làm cho chính sách truy cập mạng có hiệu lực. Điều này có nghĩa, mức độ kịp thời trong việc ngăn chặn các dòng traffic không mong muốn của firewall phụ thuộc rất lớn vào khả năng nhận định tình hình truy cập mạng để đưa ra luật mới và tốc độ cập nhật bảng luật cho firewall của người quản trị an ninh mạng. Như vậy, nếu chúng ta xây dựng được một hệ thống hỗ trợ firewall, mà nó có khả năng cao trong việc phát hiện các dòng traffic, các packet có hành động đáng ngờ và lập tức cập nhật bảng luật lọc gói cho firewall một cách tự động thì firewall sẽ làm tốt hơn nhiệm vụ của mình trong ngăn chặn kịp thời các traffic không mong muốn. Các HIDS phần mềm tự tạo có thể làm tốt nhiệm vụ hỗ trợ này.

Cả Firewall và IDS đều có thể là thiết bị phần cứng, hoặc chương trình phần mềm. Trong mô hình đề xuất của chúng tôi, firewall được chọn là công cụ mã nguồn mở IPTables. IDS là chương trình Sniffer được chúng tôi xây dựng từ ngôn ngữ Python. Đây được xem là đóng góp chính của bài báo này.

Theo đó, IPTables vừa thực hiện nhiệm vụ điều khiển truy cập mạng, vừa sẵn sàng nhận lệnh thay đổi bảng luật lọc gói từ Sniffer. Các dòng traffic, các packet sau khi qua được firewall IPTables sẽ được hướng tới Sniffer, Sniffer bắt lại và lấy ra các thông tin cần thiết từ chúng, từ đó phân tích, thống kê và dự báo, để xác định nguồn gốc của traffic, của packet có hành động đáng nghi ngờ. Và rồi thực hiện cập nhật bảng luật của IPTables một cách tự động và tức thời. Từ đây, firewall IPTables sẽ ra quyết định điều khiển truy cập theo bảng luật mới này. Đây là mục tiêu hoạt động của mô hình chúng tôi đề xuất.

Thuận lợi của các firewall mã nguồn mở, trong đó có IPTables, là ta có thể cập nhật bảng luật (chính xác là file luật lọc gói) của nó theo cơ chế dòng lệnh, điều này được thực hiện dễ dàng từ chương trình Python. Đây là một trong những lý do khiến chúng tôi nghĩ đến việc tự động hóa việc cập nhật bảng luật lọc gói cho các firewall mã nguồn mở.

2. Kiến thức liên quan

Trong phần này chúng tôi xin được dẫn lại những kiến thức cơ bản nhất, một cách cô đọng nhất, về hai hệ thống an ninh mạng không thể thiếu trong các mạng doanh nghiệp có tính an ninh cao hiện nay, đó là Firewall và IDS. Điều này là cần thiết để chúng ta dễ dàng thấy được ý nghĩa của sự kết hợp trong hoạt động giữa chúng trong mô hình mạng mà chúng tôi đề xuất ở bài báo này (ở phần III).

2.1. Nguyên lý hoạt động của Packet filtering firewall

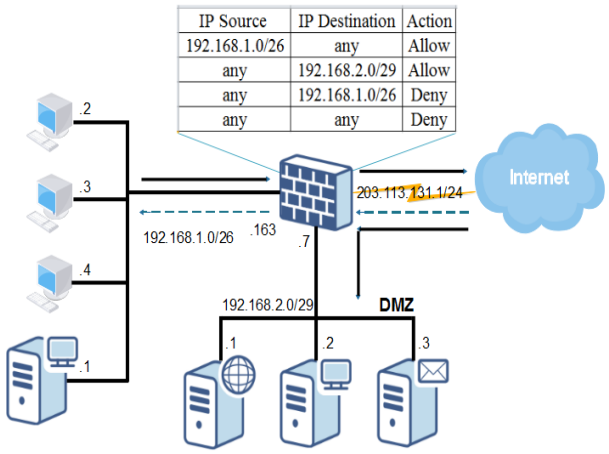
Firewall là thiết bị điều khiển truy cập mạng, nó đóng vai trò như cổng vào/ra mạng (thường gọi là gateway mạng). Theo đó, mọi dòng traffic đi vào/đi ra giữa mạng bên ngoài, thường là Internet, và mạng nội bộ của doanh nghiệp đều phải chịu sự kiểm soát và điều khiển của Firewall. Firewall dựa vào chính sách truy cập mạng, được thiết kế dưới dạng Bảng luật chính

sách, và các thông tin liên quan chứa trong dòng traffic/chứa trong gói tin - thường là IP Address, Protocols và Port number - để quyết định cho phép một traffic/một packet nào đó có được phép đi qua nó hay không.

Nếu xét về nguyên lý hoạt động thì firewall được chia làm hai loại chính. Đó là, firewall lọc gọi (Packet filtering firewall), hoạt động tương ứng tại tầng Network của mô hình mạng OSI hay TCP/IP, và firewall tầng ứng dụng (Application layer firewall), còn gọi là Proxy, hoạt động tương ứng tại tầng' Application của mô hình mạng kể trên. Các mạng doanh nghiệp hiện nay thường sử dụng loại firewall được thiết kế từ sự kết hợp nguyên lý hoạt động của hai loại firewall này.

Có thể nói, 3 thành phần chính của firewall lọc gói là: Đơn vị điều khiển việc vào/ra của các tin; Bộ lọc khảo sát gói, thường gọi là Bộ lọc gói; và Bảng luật lọc gói. Bảng luật lọc gói, được người quản trị an ninh mạng cài vào firewall, nó là cơ sở để Bộ lọc gói ra quyết định cho phép gói tin đến firewall có được đi qua nó để hướng đến đích của gói tin hay không.

Hình 2.1 là sơ đồ minh họa mạng doanh nghiệp, trong đó có sử dụng firewall lọc gói như là một gate của mạng, mọi gói tin từ mạng Internet vào mạng bên trong và từ mạng bên trong ra Internet đều phải đi qua firewall.



Hình 2.1: Sơ đồ mạng với sự xuất hiện của firewall

Từ sơ đồ mạng, ta có thể thấy được chính sách truy cập mạng của mạng doanh nghiệp này là như sau: Cho phép các gói tin từ Internet chỉ được đi vào vùng mạng 192.168.2.0/24 - vùng DMZ của mạng doanh nghiệp - không được phép đi vào vùng mạng 192.168.1.0/24, vùng mạng người dùng bên trong. Tuy nhiên, mọi dòng traffic từ vùng mạng người dùng và vùng mạng DMZ đều có thể đi ra mạng Internet. Chính sách này được người quản trị an ninh mạng chuẩn hóa thành các luật lọc gói, rồi xây dựng thành Bảng luật lọc gói (được minh họa trên sơ đồ) và sau đó là cài đặt vào firewall. Mỗi khi muốn thay đổi chính sách điều khiển truy cập mạng thì người quản trị an ninh mạng phải cập nhật lại Bảng luật lọc gói cho firewall thì chính sách mới đó mới có hiệu lực.

Khi một gói tin TCP/IP đến firewall lọc gói, Bộ lọc gói sẽ tách lấy những thông tin cần thiết trong header của gói tin này, như: Địa chỉ IP nguồn, địa chỉ IP đích, Port nguồn, Port đích và Protocols, sau đó tiến hành so khớp những thông tin có được với các luật (Rule) trong Bảng luật lọc gói (Rule Table). Khi sự "khớp" được tìm thấy tại một luật nào đó thì Bộ lọc gói sẽ ra quyết định, cho phép (Allow) gói tin đi qua hoặc từ chối (Deny) không cho gói tin đi qua, dựa vào thông tin được ghi ở cuối dòng luật này (thường được ghi tại cột Allow/Deny ở cuối Bảng luật).

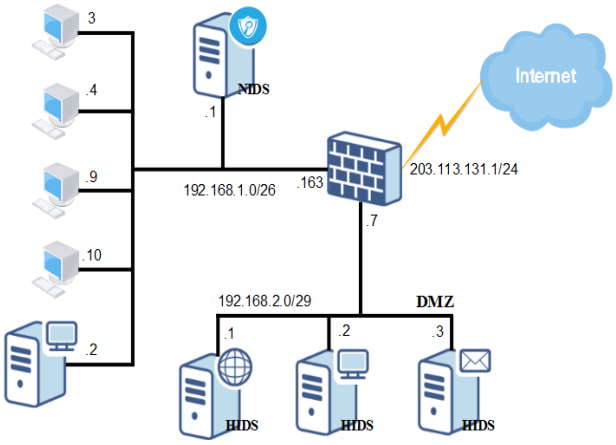
Firewall lọc gói có nhiều ưu điểm so với các loại firewall khác như tốc độ cao, dễ cài đặt và chi phí thấp. Nhưng nó cũng có nhiều nhược điểm như không hiểu các giao thức tầng ứng dụng, không phân biệt được gói tin tốt/gói tin xấu, việc tạo và cập nhật bảng luật phụ thuộc hoàn toàn vào con người và hoàn toàn không có cơ chế xác thực người dùng và nguồn gốc của các gói tin đến firewall.

Có thể thấy, hiệu quả sử dụng firewall trong việc ngăn chặn những dòng traffic, những gói tin không mong muốn đi vào/đi ra mạng nội bộ là

phụ thuộc lớn vào người quản trị an ninh mạng. Họ có nhiệm vụ tạo ra Bảng luật cho firewall và cập nhật bảng luật này khi phát hiện thấy cần phải thay đổi chính sách truy cập mạng để bảo vệ mạng. Firewall chỉ có nhiệm vụ làm cho chính sách an ninh mạng cho hiệu lực.

2.2. Nguyên lý hoạt động của Intrustion Detection System

Theo [3], IDS là hệ thống giám sát truy cập mạng, nó có thể là thiết bị phần cứng hoặc là ứng dụng phần mềm. Nó có nhiệm vụ theo dõi các dòng traffic đi vào/đi ra mạng, hoặc một máy tính trong mạng, để phát hiện và đưa ra lời cảnh báo cho các bộ phận liên quan về các truy cập bất thường, các truy cập không được phép vào hệ thống mạng hoặc vào các máy tính nào đó trong mạng.



Hình 2.2: Sơ đồ mạng với sự xuất hiện của NIDS và HIDS

Trong thực tế, IDS có thể theo dõi và phân tích mọi hoạt động của user và system, có thể thực hiện audit các tập tin system, các tập tin configuration và cả Hệ điều hành. Nó cũng có thể đánh giá sự toàn vẹn của các tập tin system và tập tin data, có thể phát hiện lỗi trong cấu hình hệ thống, nó còn có thể phát hiện và đưa ra cảnh báo nếu phát hiện hệ thống gặp nguy hiểm. Đặc biệt, IDS có thể tiến hành phân tích các mẫu (pattern) dựa trên các cuộc tấn công đã biết.

IDS có một số ưu điểm như, dễ triển khai trong một mạng doanh nghiệp sẵn có mà không ảnh hưởng đến mạng hiện tại, có thể đưa ra cảnh báo về bất thường một cách tức thời và ở nhiều dạng khác nhau. Khả năng phát hiện dòng traffic bất thường và hành động đáng ngờ của IDS có thể giúp hệ thống và quản trị viên phát hiện và ngăn chặn hiệu quả nhiều cuộc tấn công vào mạng. Quản trị viên hệ thống dễ dàng thay đổi *"cái gì cần giám sát"* của IDS thông qua việc thay đổi "*tập signature"* của nó (với loại Signature-based IDS). Ngoài ra, IDS còn có thể ghi nhật ký giám sát (log), phát hiện và báo cáo về sự thay đổi của các tập tin quan trọng chứa tại các máy tính bên trong mạng nội bộ. Tuy nhiên IDS cũng còn vài hạn chế như: IDS có thể đưa ra cảnh báo nhầm (false positive) hoặc không đưa ra cảnh báo (false negative) về sự bất thường, sự đáng ngờ đối với hệ thống, điều này có thể ảnh hưởng tiêu cực đến hoạt động bình thường của hệ thống. Khả năng phân tích các gói tin bị mã hóa của IDS cũng hạn chế, nó cũng không giúp phát hiện ra nguồn gốc của một cuộc tấn công vào mạng. IDS cũng có thể bị tấn công DoS, tấn công "đánh lừa" như những hệ thống an ninh mạng/dịch vụ mạng khác.

Chúng ta nên có sự phân biệt rõ giữa hai hệ thống an ninh mạng IDS và IPS (Intrustion Prevention System, còn gọi là Hệ thống ngăn chặn xâm nhập mạng). Cả hai đều thực hiện việc đọc và phân tích các gói tin mạng, rồi so sánh với *"tập signature"* (còn gọi là tập chữ ký, hay tập luật) về những mối đe dọa (thread), những cuộc tấn công đã biết. Trong khi IDS chỉ phát hiện và đưa ra cảnh báo về những packet bất thường thì IPS được xem là hệ thống điều khiển truy cập, nó có thể chấp nhận (accept) hoặc từ chối (reject) một packet nào đó dựa trên tập luật cho trước. IDS cần con người và/hoặc một hệ thống khác xem kết quả và đưa ra hành động xử lý tiếp theo, trong khi đó IPS chỉ cần tập luật về

các mối đe dọa đã biết được cập nhật thường xuyên, để bổ sung các mối đe dọa mới.

Trong mô hình mạng doanh nhiệp, IDS có thể đặt giữa firewall và router (kết nối Internet): Khi cần IDS giám sát traffic cả đi vào và đi ra trên toàn mạng; có thể đặt trong vùng DMZ: Khi chỉ cần IDS giám sát traffic đi vào vùng DMZ; hoặc có thể đặt sau firewall (ngay trước mạng bên trong): Khi cần IDS giám sát traffic giữa mạng bên trong với Internet và với vùng DMZ. Thông thường các HIDS được đặt ngay tại mỗi server trong vùng DMZ để giám sát và cảnh báo các truy cập bất thường, các hành động đáng ngờ vào các server này.

Một cách chung nhất IDS được chia thành 5 loại: Host-based IDS (HIDS), Network-based IDS (NIDS), Protocol-based IDS (PIDS), Application Protocol-based IDS (APIDS) và Hyber IDS (HyIDS). Nếu xét về vị trí của IDS trong mạng hay phạm vi, trên toàn mạng hay chỉ trên một máy tính, giám sát dòng traffic của IDS thì nó được chia thành 2 loại chính là HIDS và NIDS.

- NIDS: Loại IDS này thường được đặt tại một vị trí xác định nào đó trong mạng mà tại đó nó có thể theo dõi tất cả dòng traffic đi vào/đi ra tất cả các thiết bị trong mạng. NIDS quan sát và phân tích một cách liên tục mọi dòng traffic trên toàn bộ vùng mạng của nó, rồi so khớp với tập signature về các mối đe dọa, về các cuộc tấn công đã biết. Một khi có một sự "khớp" được xác định hoặc một sự bất thường được nhận ra thì NIDS sẽ gửi lời cảnh báo về những hành động bất thường này đến các hệ thống liên quan, đặc biệt, là đến người quản trị an ninh của hệ thống mạng.

- HIDS: Loại IDS này được sử dụng để theo dõi các gói tin đi vào/đi ra các host riêng lẻ trong mạng. HIDS thường được đặt ngay tại các máy tính, các thiết bị mà nó nhận nhiệm vụ theo dõi. Khi phát hiện thấy các hành động đáng ngờ, các hành động nguy hại với "thân chủ" của nó thì HIDS sẽ gửi cảnh báo đến các hệ thống liên quan hoặc đến người quản trị hệ thống để họ có hướng xử lý tiếp theo. HIDS còn được yêu cầu để theo dõi các tập tin quan trọng trên host mà nó đang thực hiện nhiệm vụ giám sát, nếu phát hiện thấy các tập tin này bị thay đổi hoặc bị xóa thì nó phải gửi cảnh báo đến các hệ thống liên quan.

Nếu dựa vào phương pháp phát hiện xâm nhập (hoạt động) của IDS thì nó gồm loại: Signature-based IDS (IDS dựa trên "tập signature") và Anomoly-based IDS (IDS dựa trên sự "bất thường").

- Signature-based IDS: IDS loại này dựa vào tập signature (chữ ký) cho trước - đây chính là cơ sở dữ liệu về các mối đe dọa và các cuộc tấn công đã biết - để nhận định dòng traffic nào có biểu hiện bất thường hay có những hành động đáng ngờ, bằng cách, "so khớp" dữ liệu phân tích được từ các dòng traffic đến với dữ liệu ghi trong tập signature. Loại IDS này có chức năng tương tự các phần mềm Antivirus, nó được sử dụng khá phổ biến và hiệu quả. Nhưng sự hiệu quả của IDS lại phụ thuộc vào tập signature, vì thế nó hoàn toàn bị động trước những "bất thường mới" và những loại "tấn công mới". Ngoài ra, nếu tập signature lớn cũng sẽ ảnh hưởng đến băng thông của mạng.

- Anomoly-based IDS: IDS loại này được sử dụng để quan sát những dòng traffic mà ẩn chứa trong đó những "bất thường mới", những loại "tấn công mới". Nó quan sát những dòng traffic được chỉ định để tìm ra "sự bình thường", đây là một trong những cơ sở để IDS phát hiện ra sự bất thường. Trong thực tế, phương pháp này đòi hỏi IDS phải có "trí tuệ" cao thì mới có thể hoàn thành tốt nhiệm vụ phát hiện về những traffic, những packet bất thường mà hệ thống chưa được biết trước đó. Hiện nay, để có được một IDS có "trí tuệ" cao thì nó phải được xây dựng theo hướng tiếp cận Machine learning.

Có thể thấy, khác nhau cơ bản giữa hai loại IDS này là, signature-based IDS dựa vào tập signature đã biết để phát hiện sự bất thường, trong khi đó, anomaly-based IDS, dựa vào những traffic "bình thường" làm cơ sở để phát hiện ra sự bất thường.

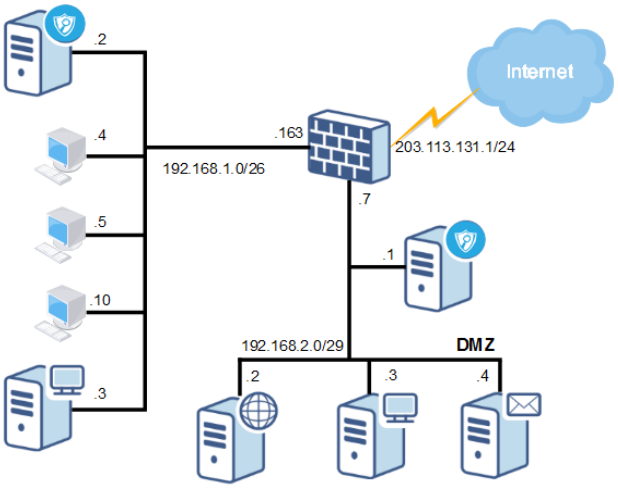
Hình 2.2 là sơ đồ minh họa mạng doanh nghiệp, trong đó sử dụng cả NIDS và HIDS. Trong mạng này, các HIDS được đặt tại các server trong vùng DMZ (Web, File, Mail), để phát hiện sự truy cập bất hợp pháp hoặc bất thường có thể đến với các server này. NIDS được đặt sau firewall để quan sát tất cả dòng traffic đi vào/đi ra mạng bên trong.

Hoạt động của loại Signature-based IDS là như sau, khi có một dòng traffic mạng đi qua nó, IDS sẽ copy traffic này lại, rồi tách lấy những thông tin cần thiết trong header và/hoặc payload từ những traffic này. Sau đó IDS tiến hành "so khớp" những thông tin này với cơ sở dữ liệu chứa tập signature về những mối đe dọa biết trước. Nếu một sự "khớp" được tìm thấy thì IDS sẽ gửi cảnh báo đến các hệ thống liên quan. Khi IDS nhận định rằng có một sự bất thường mới hay một mối đe dọa mới từ những dòng traffic qua nó thì cơ sở dữ liệu này cũng sẽ được cập nhật để chuẩn bị cho những lần "so khớp" sau này. Với loại Anomaly-based IDS, đầu tiên nó quan sát tất cả dòng traffic đi qua, thu thập thông tin liên quan để tìm ra "sự bình thường". Sau đó dựa vào "sự bình thường" này để phát hiện ra những dòng traffic ẩn chứa trong đó "sự bất thường". Loại IDS này mang lại hiệu quả cao trong việc phát hiện những mối đe dọa mới, những cuộc "tấn công mới". Tuy nhiên, IDS phải có "trí tuệ" thì mới làm tốt việc này.

3. Mô hình đề xuất

3.1. Mô hình mạng

Mô hình đề xuất về sự kết hợp hoạt động giữa firewall và HIDS của chúng tôi được thể hiện ở Hình 3.1 sau đây:



Hình 3.1 Sơ đồ mạng với sự kết hợp giữa firewall và HIDS

Trong mô hình này:

• Firewall chịu trách nhiệm điều khiển truy cập cho toàn mạng. Về cơ bản, firewall chỉ cho phép các kết nối từ Internet vào vùng DMZ, cấm các kết nối từ Internet vào vùng mạng người dùng bên trong. Nhưng mọi kết nối từ bên trong mạng đi ra thì đều được phép. Chúng tôi sử dụng phần mềm tường lửa mã nguồn mở IPTables để xây dựng firewall này. Bảng luật lọc gói ban đầu được xây dựng thông qua các lệnh cho phép của IPTables.

• IDS chịu trách nhiệm quan sát dòng traffic đi vào vùng DMZ. Nếu IDS phát hiện thấy sự bất thường hay hành động đáng ngờ từ dòng traffic hay packet nào đó thì nó sẽ báo cáo điều này đến các thành phần/đối tượng liên quan. Đặc biệt, Sniffer có thể tự động cập nhật bảng luật lọc gói trên IPTables. Chúng tôi sử dụng thư viện Scapy của ngôn ngữ Python để xây dựng chương trình Sniffer thực hiện chức năng và nhiệm vụ này của IDS.

Điểm mới của mô hình đề xuất này là chúng tôi đã tạo được kết nối trong hoạt động và thực hiện nhiệm vụ được giao giữa chương trình Sniffer với tường lửa mã nguồn mở IPTables: Một khi đã xác định được nguồn gốc của dòng traffic hoặc packet được cho là bất thường hay được cho là có hành động đáng ngờ thì Sniffer

lập tức gửi lệnh đến IPTables để cập nhật lại bảng luật lọc gói của nó. Firewall IPTables sẽ hoạt động theo bảng luật lọc gói mới này để ngăn chặn những dòng traffic, những gói tin không mong muốn đi vào mạng.

Ưu điểm của mô hình đề xuất là rất rõ, ngoài chức năng thông thường của một IDS chuyên dụng, Sniffer hoàn toàn chủ động trong việc thay đổi bảng luật lọc gói trên tường lửa IPTables một cách tức thời và tự động. Điều này không những giúp kịp thời ngăn chặn những ý đồ phá hoại mạng của tổ chức từ Internet mà còn giúp giảm công sức của người quản trị an ninh mạng.

```
def update_rule1(black,Port):
        cmd = 'iptables -A FORWARD -i
ens33 -o
        ens38 -p tcp -d 192.168.10.50 --dport
        "'+Port+'" -m string --algo bm --string
        "'+black+'" -j DROP'
        p = paramiko.SSHClient()
        p.set_missing_host_key_policy
                        (paramiko.
AutoAddPolicy())      p.connect(add1,
username=user1,
                         password=pass1)
          stdin, stdout, stderr =
                        p.exec_command(cmd)
          print
"……................................", Port
#-------------------------------------------------
def update_rule2 (ipsrc):
          cmd = 'iptables -A FORWARD -i
ens33 -o
                    ens38 -p tcp -s '+ipsrc+'
                     -d address2 -j DROP'
          p = paramiko.SSHClient()
          p.set_missing_host_key_policy
                        (paramiko.
AutoAddPolicy())      p.connect(add2,
username=user2,
                         password=pass2)
          stdin, stdout, stderr =
p.exec_command(cmd)
```

Hình 3.2 Đoạn code chính của chương trình

3.2. Công việc liên quan

Đầu tiên chúng tôi thực hiện các công việc cần thiết để đưa mạng, theo thiết kế, của mô hình đề xuất đi vào hoạt động. Chúng tôi tiến hành xây dựng Bảng luật lọc gói của tường lửa IPTables theo chính sách truy cập mạng đã đưa ra, trong đó, chúng tôi hướng mọi dòng traffic sau khi đã vượt qua được tường lửa IPTables thì phải đi qua IDS, máy chạy chương trình Sniffer.

Đóng góp chính của chúng tôi ở đây không chỉ là đề xuất mô hình mới trong sự kết hợp giữa IDS và Firewall mà còn là xây dựng được chương trình Sniffer từ Python. Chương trình Sniffer gồm các function chính sau đây:

- Function *process_packet* (): Đây là function khá quan trọng của Sniffer, nó có nhiệm vụ sao chép tất cả các packet đi qua các port được chỉ định (đã được chỉ ra ở các tập tin: *Listports và Blacklist*). Sau đó tách lấy các thông tin cần thiết từ header của packet này như: Loại gói tin (TCP hay UDP), địa chỉ IP nguồn, Port nguồn… Dữ liệu này sẽ là đầu vào cho function *test_for_active*.
- Function *test_for_active ()*: Thực hiện chức năng so khớp thông tin của các packet nhận được từ *process_packet* với thông tin lưu trong *Listports* và *Blacklist.* Nếu tìm thấy sự khớp từ một traffic hay packet nào đó thì Sniffer tiến hành đếm sự xuất hiện của chúng, nếu số lần này chạm ngưỡng đã được xác định thì *test_for_active* sẽ tiến hành cập nhật bảng luật của IPTables để firewall này ngăn chặn các traffic này lại, không cho đi vào mạng. Function này còn có nhiệm vụ phát hiện và cảnh báo những về dòng traffic đáng ngờ, đó thường là những dòng traffic được gửi đến Sniffer một cách ồ ạt từ một địa chỉ IP hoặc từ nhiều địa chỉ IP.
- Function *setup_logfile ()*: Thiết lập các thông số liên quan đến tập tin nhật ký (logfile) mà Sniffer sử dụng để lưu thông tin về những

traffic, packet mà nó được chỉ định cần phải chú ý trong quá trình quan sát.

Ngoài ra, chúng tôi còn xây dựng 1 class và 2 chương trình hỗ trợ:

• Class *limitedPool*: Class này gồm function *init_* và function *_setup_queues*. Function *init_* thực hiện việc khởi tạo các hàng đợi cho các ThreadPool, được sử dụng trong trường hợp xử lý đa luồng. Ở đây chúng tôi chọn kích thước hàng đợi tối đa là 10000 (chứa được 10000 thread). Function *_setup_queues* được sử dụng để thiết lập các thông số về hàng đợi.

Nhờ sự hỗ trợ này của Python [4] mà Sniffer có thể xử lý được tất cả packet tới nó, cho dù có đến 10000 packet được gửi đến đồng thời.

• Chương trình *update_rule_iptables.py:* Được xây dựng dựa trên giao thức SSH, được Sniffer sử dụng để gửi lệnh đến IPTables, yêu cầu tường lửa này cập nhật bảng luật để chặn các packet có chứa các string mà Sniffer cho rằng nó có nguy cơ làm tổn hại đến hệ thống mạng bên trong.

Hai function này được Sniffer sử dụng để gửi lệnh thay đổi bảng luật của IPTables, để IPTables kịp thời ngăn chặn những dòng traffic, những packet không mong muốn đi vào mạng bên trong.

• Và chương trình *report2email.py:* Được xây dựng trên nền giao thức SMTP, được sử dụng để Sniffer gửi email với nội dung về cảnh báo bất thường đến cho người quản trị an ninh mạng.

3.3. Kiểm thử và đánh giá mô hình đề xuất

➢ Chúng tôi đã tiến hành kiểm thử mô hình đề xuất theo 3 kịch bản như sau:

• ***Kiểm thử lọc string (kịch bản 1):*** Kịch bản này nhằm kiểm tra sự phối hợp giữa Sniffer với IPTables trong việc cảnh báo và ngăn chặn những dòng packet mà trong đó có chứa một string thuộc diện cần lưu ý và đã được chỉ ra ở *blacklist*.

Kết quả đúng như mong đợi, trong một khoảng thời gian xác định, nếu có lượng lớn packet đi vào Sniffer mà payload của nó có chứa cùng một string đã ghi trong *blacklist* thì ngay lập tức những packet này sẽ bị chặn bởi IPTables.

• ***Kiểm thử lọc theo IP nguồn (kịch bản 2):*** Kịch bản này nhằm kiểm tra phản ứng của Sniffer khi có quá nhiều kết nối đến từ một nguồn có địa chỉ IP xác định. Sniffer phải xác định được đây là kết nối đáng ngờ và IPTables phải chặn được những traffic xuất phát từ địa chỉ IP này.

Kết quả là, trong một khoảng thời gian xác định, nếu có lượng lớn traffic đi vào Sniffer mà có cùng một địa chỉ IP nguồn thì ngay lập tức dòng traffic này sẽ bị chặn bởi IPTables.

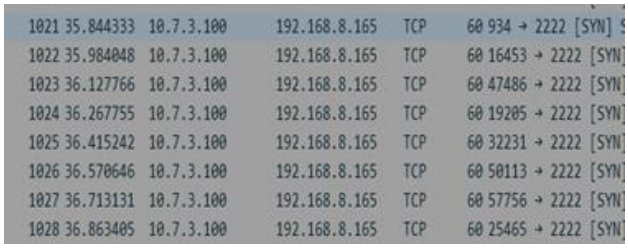
• ***Kiểm thử ngăn chặn tấn công SYN_Flood (kịch bản 3):*** Kịch bản này yêu cầu Sniffer phải phản ứng khi có quá nhiều gói tin TCP_SYN gửi đến nó (nghi ngờ có tấn công DoS dạng SYN_Flood). Trong tình huống này Sniffer sẽ phải gửi yêu cầu đến IPTables, nhờ nó chặn các dòng traffic chứa gói tin TCP_SYN, công việc còn lại là của IPTables.

Không ngoài dự đoán, trong một khoảng thời gian xác định, nếu có lượng lớn packet TCP_SYN đi vào Sniffer thì lập tức dòng traffic có packet khởi tạo này bị chặn bởi IPTables.

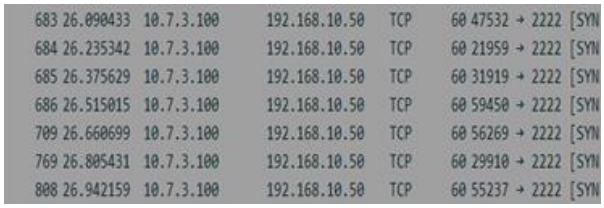
➢ Trong quá trình kiểm thử, chúng tôi cũng đã tiến hành đánh giá hiệu quả phối hợp hoạt động giữa Sniffer và IPTables. Ở đây, chúng tôi sử dụng công cụ Tcpdump và Wireshark để kiểm tra kết quả lọc và chặn packet của mô hình đề xuất:

• ***Đối với việc lọc string:*** Trong khoảng thời gian xác định chúng tôi tấn công vào IPTable bằng cách tạo và gửi đến IPTables khoảng hơn 1000 packet, trong đó có một lượng lớn vượt ngưỡng cho phép, packet có chứa string đã được yêu cầu Sniffer theo dõi (ở đây là lệnh "ls –ls"). Chúng tôi tiến hành đồng thời dump các packet trên firewall và trên server chạy Sniffer. Sau đó tiến hành phân tích hai kết quả dump được thì

thấy rằng: Có hơn 1000 packet đến firewall nhưng chỉ có hơn 800 packet đến được server.



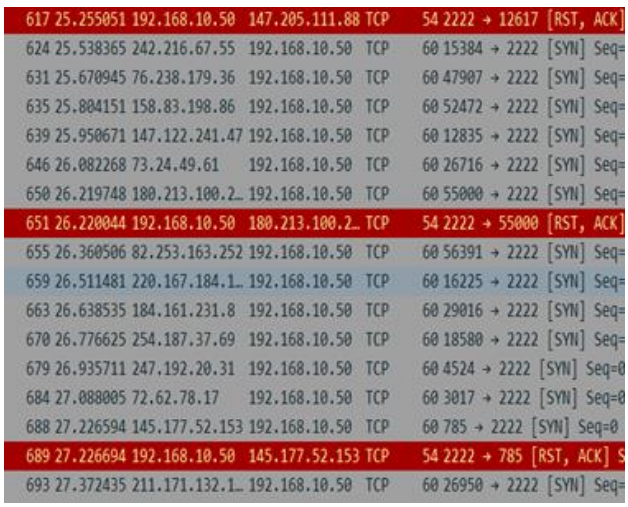
Hình 3.3a Các packet nhận được trên tường lửa IPTables



Hình 3.3b Các packet nhận được trên server Sniffer

Như vậy, Tcpdump đã cho thấy, trước sự quan sát và hành động của Sniffer và IPTables thì các packet (một lượng lớn) có chứa string "ls – ls" đã bị chặn lại ở IPTable.

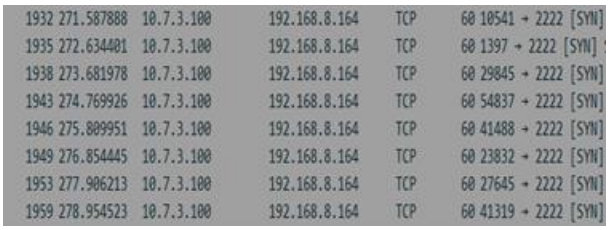
- ***Đối với việc ngăn chặn tấn công SYN_Flood:*** Chúng tôi thực hiện tấn công vào mạng đề xuất bằng cách tạo và gửi một lượng lớn packet TCP_SYN từ bên ngoài, qua IPTables, đến server Sniffer. Sau đó chúng tôi tiến hành dump các packet đến/đi ở Sniffer. Kết quả dump được nhìn thấy trên Tcpdump như sau:



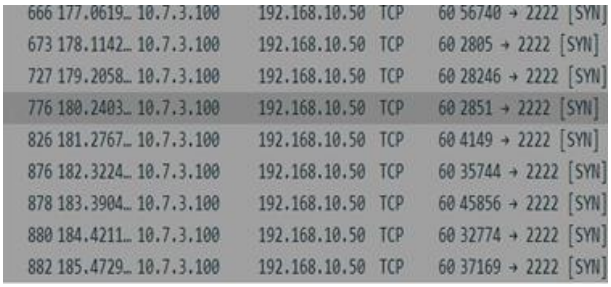
Hình 3.4 Các packet TCP_SYN và RST, ACK

Trong quá trình "bắt tay 3 bước" TCP, thông thường, cứ mỗi khi bên đích nhận được một packet [TCP_SYN] thì nó phải gửi ngay một packet phản hồi [TCP_RST, ACK]. Nhưng ở đây, do nghi ngờ hệ thống bị tấn công DoS theo dạng SYN_Flood nên Sniffer đã yêu cầu IPTables hạn chế hồi đáp các packet khởi tạo kết nối [TCP_SYN]. Điều này đã được kiểm chứng bởi Tcpdump.

- ***Đối với việc lọc theo IP nguồn:*** Thử nghiệm này được tiến hành tương tự như *"lọc string"* nhưng ở đây yêu cầu Sniffer phát hiện và ngăn chặn sự bất thường đến từ những dòng traffic đến từ cùng một địa chỉ IP nguồn. Kết quả dump và phân tích các packet nhận được tại IPTables và Sniffer được thể hiện ở hai hình sau:



Hình 3.5a Các packet nhận được trên tường lửa IPTables



Hình 3.5b Các packet nhận được trên server Sniffer

Như vậy, Tcpdump đã cho thấy, nhờ sự quan sát và hành động của Sniffer và IPTables mà các packet xuất phát từ IP: 10.7.7.100 (với lượng lớn) đã bị chặn lại ở IPTables.

Hầu hết các sản phẩm firewall mã nguồn mở đều cho phép tương tác với nó thông qua cơ chế dòng lệnh, nên việc chúng tôi sử dụng IPTables trong các kịch bản thử nghiệm và đánh giá của mình vẫn không làm mất tính tổng quát của mô hình đề xuất.

Từ kết quả đánh giá, chúng tôi tin tưởng rằng hệ thống đề xuất hoàn toàn có thể đóng vai trò phát hiện và xâm nhập trái phép vào mạng nội bộ doanh nghiệp như các hệ thống IDS và firewall riêng lẻ khác trên thị trường sản phẩm an toàn thông tin.

4. Kết luận

Trong bài báo này, chúng tôi đề xuất và đã triển khai thành công một mô hình mới trong việc kết hợp hoạt động giữa IDS và firewall mã nguồn mở IPTables. Chúng tôi cũng đã xây dựng được chương trình Sniffer đảm nhận vai trò IDS của hệ thống mạng. Sự cập nhật bảng luật này là hoàn toàn tự động nên có tính tức thời rất cao.

Tính chính xác trong việc đưa ra nhận định về các hành động đáng ngờ của một IDS vừa phụ thuộc vào việc thu thập và phân tích thông tin, từ những dòng traffic vào/ra mạng, vừa phụ thuộc vào khả năng thống kê và dự báo của nó. Có thể nói, "trí tuệ" của bộ phận này quyết định rất lớn đến độ chính xác trong việc nhận định dòng traffic nào là có hành động đáng ngờ trong số rất nhiều dòng traffic đi qua IDS. Cũng vậy, với khả năng "giả mạo" và "đánh lừa" ngày càng cao của hacker trên không gian mạng, nếu không có "trí tuệ" cao thì IDS khó có thể xác định chính xác một packet nào đó là có mang theo mã độc hay không. Hạn chế của Sniffer của chúng tôi là "trí tuệ" của nó còn thấp. Trong tương lai chúng tôi sẽ cải thiện "trí tuệ" của Sniffer bằng cách xây dựng nó theo hướng tiếp cận Machine learning [5].

Tài liệu tham khảo

[1] Kanika & Urmila, *"Security of Network Using IDS and Firewall",* International Journal of Scientific and Research Publications, vol. 3, iss. 6, June 2013.

[2] Waleed Bul'ajoul, Anne James & Mandeep Pannu, *"Improving network intrusion detection system performance through quality of service configuration and parallel technology*", Journal of Computer and System Sciences, pp. 981-999, vol. 81, iss. 6, September 2015.

[3] D. Ashok Kumar & S.R Venugopala, "*Intrusion Detection Systems: A review*", International Journal of Advanced Research in Computer Science, vol. 8 , no. 8, Octorber 2017.

[4] Mrinal Wahal, Tanupriya Choudhury & Manik Arora, "*Intrusion Detection System in Python"*, 8th International Conference on Cloud Computing, Data Science & Engineering, pp. 348-353, August 2018.

[5] Suad Mohamed Othman, Fadl Mutaher Ba-Alwi, Nabeel T.Alsohybe & Amal Y.Al-Hashida,*"Intrusion detection model using machine learning algorithm on Big Data environment"*, Journal of Big Data, no. 34, September 2018.