Phát hiện lọc bỏ nhanh các gói tin giả mạo trong tấn công mạng TCP SYN Flood

Trần Mạnh Thắng, Nguyễn Khanh Văn
Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách khoa Hà Nội
E-mail: thang197@gmail.com, vannk@soict.hust.edu.vn
Tác giả liên hệ: Trần Mạnh Thắng
Ngày nhận: 29/05/2017, ngày sửa chữa: 05/07/2017, ngày duyệt đăng: 01/09/2017

Tóm tắt: Tấn công từ chối dịch vụ phân tán (DDoS) theo hình thức gửi tràn ngập gói khởi tạo kết nối (gói tin SYN) là một trong những dạng tấn công mạng rất nguy hiểm và khó phòng-chống. Bởi vì với dạng tấn công này, tin tặc cố tình gửi gói tin khởi tạo kết nối giả mạo địa chỉ nguồn mà đảm bảo các trường thông tin như của gói tin khởi tạo bình thường. Đã có nhiều kết quả nghiên cứu và giải pháp đề xuất về việc phát hiện và lọc bỏ những gói tin giả mạo này, nhưng phần lớn các xử lý yêu cầu nhiều tài nguyên hệ thống để xử lý, do đó hiệu quả sẽ bị ảnh hưởng nhiều khi luồng tấn công DDoS lớn. Trong nghiên cứu này, chúng tôi đề xuất một thuật toán phát hiện và lọc bỏ nhanh các gói tin giả mạo mà sử dụng rất ít tài nguyên hệ thống và đạt tỷ lệ phát hiện rất cao.

Từ khóa: *Tấn công từ chối dịch vụ phân tán, tấn công SYN Flood.*

Title: **A Method for Fast Filtering SYN Flood Spoof Source in DDoS Attack**

Abstract: A distributed denial-of-service attack characterized by flood SYN packets is one of network attacks to make the information system unavailable. This attack becomes more dangerous and difficult to prevent and defend when attackers try to send flood SYN packets with spoof sources. Especially, the packets have the information field as that of normal SYN packets. Most of current methods require a training phase to build normal parameters, which will be used in the detection phase to detect spoof packets. In this research, we propose a method that can detect and filter spoof packets quickly without using the training phase. This method requires very few resources and the detection rate is very high.

Keywords: *DDoS, TCP SYN Flood.*

I. GIỚI THIỆU

Tổng quan về tấn công TCP SYN Flood. Tấn công từ chối dịch vụ (DoS: Denial-of-Service) là dạng tấn công mạng phổ biến và nguy hiểm bậc nhất. Trong đó, kẻ tấn công muốn làm cho hệ thống bị quá tải và không thể phục vụ các yêu cầu hợp lệ do cạn kiệt tài nguyên (năng lực tính toán, bộ nhớ, v.v.). Tấn công từ chối dịch phân tán (DDoS: Distributed DoS) là dạng đặc biệt của tấn công DoS. Khi sử dụng DDoS, tin tặc tìm cách xây dựng mạng botnet (một số lượng lớn các máy tính trên mạng bị điều khiển bởi một máy chủ điều khiển C&C) để thực hiện tấn công DDoS thay vì chỉ thực hiện tấn công từ một máy như dạng tấn công DoS thông thường. Mạng botnet được xây dựng bằng nhiều hình thức khác nhau. Một hình thức phổ biến là tin tặc tạo ra các mã độc và cho lây lan trên môi trường mạng. Mỗi máy tính bị nhiễm mã độc sẽ trở thành thành viên của mạng botnet (bot) và chịu sự điều khiển của máy chủ C&C để thực hiện các cuộc tấn công DDoS.

Có nhiều hình thức tấn công DDoS khác nhau, như sau.

Ping Flood: Dạng tấn công DDoS bằng cách gửi tràn ngập gói tin ICMP hoặc UDP có kích thước lớn.

Teardrop: Dạng tấn công DDoS bằng cách gửi tràn ngập các gói tin phân mảnh có offset trùng nhau hoặc ngắt quãng là phía máy nhận gói tin không thể xử lý được gây làm treo hệ thống.

TCP Syn Flood: Đối với dạng tấn công này, tin tặc gửi tràn ngập gói tin SYN nhưng giả mạo địa chỉ IP nguồn [1] thông qua mạng botnet (sau đây gọi là gói tin giả mạo) là một trong những hình thức tấn công khó phòng, chống nhất. Do tin tặc cố tình tạo ra các gói tin giả mạo có các trường thông tin như gói tin bình thường nên máy chủ nạn nhân không thể phân biệt được các gói tin giả mạo này nên nhanh chóng bị cạn kiệt tài nguyên.

Đối với dạng tấn công TCP SYN Flood, tin tặc khai thác điểm yếu của giao thức phổ biến TCP (giao thức giao vận trong kiến thúc TCP/IP của Internet) để tấn công DDoS.

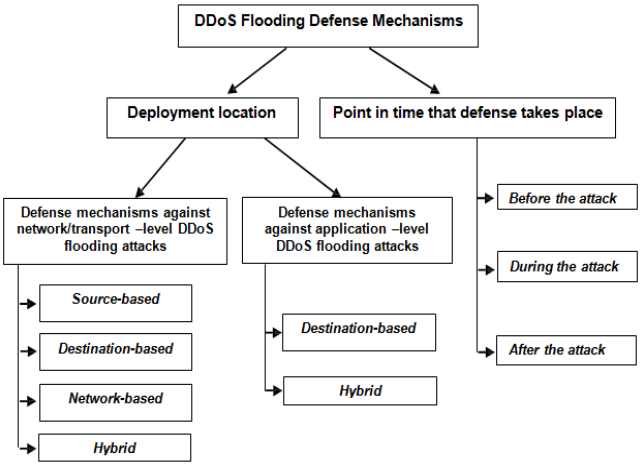
Cụ thể, trong quá trình bắt tay 3 bước (3-way handshake sub-protocol) gói tin SYN được sử dụng để khởi tạo một kết nối TCP. Các gói tin này sẽ được gửi tràn ngập tới máy chủ với địa chỉ IP nguồn giả mạo. Máy chủ phải dành tài nguyên để phục vụ các kết nối đối với các gói tin giả mạo, dẫn tới cạn kiệt tài nguyên.

Ý tưởng cơ bản của giải pháp. Các gói tin giả mạo cũng không khác gì bình thường nếu xét riêng từng gói tin. Tuy nhiên, chúng tôi cho rằng, có những mối liên hệ riêng giữa chúng mà có thể coi là dấu hiệu tìm vết, khi chúng ta quan sát cả một luồng các gói tin giả mạo phát đi từ một máy tính. Dựa trên dấu hiệu này, ta có thể xây dựng các giải pháp để phát hiện và thực hiện lọc bỏ (nhiều nhất có thể được) các gói tin giả mạo sử dụng trong tấn công TCP SYN Flood. Trong nghiên cứu này, chúng tôi phát hiện ra rằng khi một máy tính gửi ra một gói tin mà không phân biệt địa chỉ IP đích hay dịch vụ sử dụng thì giá trị IPD (trường Identification trong IP Header [2]) sẽ tăng lên 1 đơn vị. Điều này có nghĩa khi quan sát ở phía máy chủ thì ta sẽ nhận được chuỗi các gói tin có giá trị PID tăng liên tục, nếu chúng gửi đi từ một máy. Chúng tôi mô tả cụ thể tính chất này tại mục III-1.

Trong một bài báo trước [3], chúng tôi đã đề xuất phương pháp dùng thuật toán DBSCAN để nhóm các gói tin SYN có địa chỉ IP khác nhau nhưng có giá trị PID tăng dần vào một cụm giá trị liên tục (Cluster). Với mỗi cụm[1], chúng tôi xác định được giá trị trường PID của gói tin tấn công tiếp tục nếu có (Expected PID - E_{PID}) sẽ thuộc về cụm này. Đối với phương pháp này, chúng tôi phải thực hiện quá trình máy học (training phase) để tìm ra các cụm và giá trị E_{PID} cho mỗi cụm.

Trong bài báo này, chúng tôi đề xuất một phương án mới cho phép tìm ra các gói tin có giá trị PID tăng liên tục với cơ chế thuật toán hoàn toàn khác (không dựa trên quá trình máy học như trước đây), trực tiếp phát hiện ra các gói tin tấn công dựa vào quan sát về PID nói trên (và một số quan sát khác) và một cấu trúc dữ liệu tổ chức đặc biệt để lưu trữ và đối sánh thông tin. Phương pháp này cũng cho phép tự loại bỏ các gói tin ngay khi phát hiện gói đầu tiên trong chuỗi gói tin giả mạo gửi đến hệ thống, cũng như cho phép những gói tin thực kết nối đến hệ thống trước khi xảy ra quá trình TCP retransmission - những ưu điểm mới so với phương án trước.

Bài báo có cấu trúc như sau: Mục I giới thiệu tổng quan về tấn công TCP SYN Flood và ý tưởng cơ bản của giải pháp đề xuất. Mục II giới thiệu tổng quan lĩnh vực về các nghiên cứu có liên quan. Mục III đưa ra các cơ sở lý thuyết của phương pháp đề xuất. Mục IV trình bày về cơ chế thuật toán loại bỏ nhanh gói tin giả mạo. Mục V đưa ra kết quả đánh giá thực nghiệm. Mục VI đưa ra kết luật và hướng phát triển tiếp theo.



Hình 1. Phân loại phương pháp phòng chống tấn công DDoS.

II. TỔNG QUAN LĨNH VỰC

Theo nghiên cứu tổng quan về tấn công DDoS trong [4], về cơ bản các giải pháp phòng-chống DDoS có thể phân loại theo hai yếu tố: vị trí triển khai giải pháp và thời điểm xử lý tấn công DDoS. Cách phân loại này được thể hiện bằng sơ đồ ở Hình 1[2].

Với yếu tố đầu, ta có thể thấy có hai lớp giải pháp: (A1) lớp các giải pháp phòng chống cho các dạng tấn công nhằm vào các tầng giao vận hoặc thấp hơn trong mô hình OSI [5]; (A2) lớp các giải pháp phòng chống cho các dạng tấn công nhằm vào tầng ứng dụng. Với yếu tố thứ hai, ta có thể phân loại thành các lớp giải pháp nhằm vào ba giai đoạn phân biệt: giai đoạn trước khi xảy ra tấn công (phát hiện nguy cơ), giai đoạn đang xảy ra tấn công (phát hiện sự kiện) và giai đoạn sau khi bị tấn công (khắc phục hậu quả và truy cứu).

Nhóm giải pháp (A1) nói trên lại có thể phân loại thành bốn nhóm: Nhóm giải pháp áp dụng ở gần nguồn tấn công (source-based); Nhóm áp dụng ở phía đối tượng được bảo vệ (destination-based); Nhóm áp dụng ở hạ tầng mạng trung gian (network-based) và Nhóm giải pháp kết hợp (hybrid). Nhóm (A2) được chia làm hai nhóm: nhóm giải pháp áp dụng phía đối tượng được bảo vệ và nhóm giải pháp kết hợp.

Trong bài báo này, chúng tôi tập trung nghiên cứu phương pháp phòng-chống tấn công thường được gọi là SYN Flood đã đề cập, trong đó các gói tin SYN đều mang địa chỉ IP

[1]Điều kiện để tạo thành một cụm là khi hệ thống nhận được 03 gói tin có PID tăng liên tục, nhưng có địa chỉ IP khác nhau. Các gói tin nằm trong cụm được coi là các gói tin giả mạo gửi đến hệ thống từ cùng một máy tấn công và do đó, gói tin tiếp theo có giá trị PID trùng với E_{PID} sẽ được cho là gói tin giả mạo tiếp theo gửi từ cùng máy tấn công này.

[2]Trong sơ đồ này, chúng tôi sử dụng thuật ngữ tiếng Anh để bạn đọc tiện đối chiếu với tài liệu quốc tế.

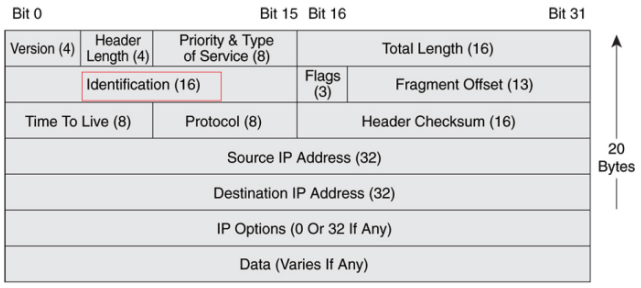
nguồn giả mạo. Giải pháp của chúng tôi thuộc dạng cho đối tượng được bảo vệ (destination-based) trong lớp (A1) nói trên. Sau đây, chúng tôi sẽ đi sâu phân tích một số nghiên cứu liên quan như sau.

Cơ chế IP Traceback [6, 7]: Phương pháp này đề xuất cơ chế truy vết kẻ tấn công dựa trên việc lưu lại thông tin về đường đi từ nguồn đến đích (trong trường thông tin Packet Identification của gói tin) ứng với mỗi địa chỉ IP nguồn. Dựa vào thông tin đường đi này (con đường phổ biến kết nối từ một IP nguồn nào đó) người ta có thể phát hiện ra các gói tin giả mạo: gói tin giả mạo là gói tin có thông tin đường đi không khớp với xu hướng phổ biến đã phát hiện trước đó (đối với một địa chỉ IP nguồn). Tuy nhiên, phương pháp này có hạn chế lớn là yêu cầu các trạm router trung gian phải hỗ trợ cơ chế đánh dấu đường đi nên rất khó triển khai trong thực tế.

Phương pháp Management Information Base (MIB) [8–10]: Phương pháp này kết hợp các thông tin có trong mỗi gói tin và thông tin định tuyến để phát hiện tấn công DDoS. Phương pháp này đề xuất so sánh thông tin trong mỗi gói ICMP, TCP, UDP (khi tấn công có thể đang xảy ra) với những thông tin tương ứng đã được phân tích và lưu trữ trong khi chưa xảy ra tấn công để tìm ra những gói tin có thông tin khác thường.

Các cơ chế Packet marking and filtering (Đánh dấu và lọc bỏ gói tin): Một nghiên cứu đề xuất cơ chế *History-based IP filtering* [11], trong đó người ta lưu lại thông tin các địa chỉ IP nguồn thường xuyên kết nối vào hệ thống khi tấn công DDoS chưa xảy ra; trong tấn công, các địa chỉ IP đã lưu lại này sẽ được kết nối vào hệ thống, còn lại sẽ bị lọc bỏ. Phương pháp *Hop-count filtering* [12] thì chú ý ghi nhận thông tin hop-count tương ứng với mỗi IP nguồn (qua các gói tin được lưu lại khi tấn công DDoS chưa xảy ra). Khi tấn công DDoS xảy ra, các gói tin có thông tin hop-count khác nhiều với thông tin đã lưu trước đó ứng với địa chỉ IP nguồn tương ứng sẽ được coi là giả mạo và bị lọc bỏ. Phương pháp *Path Identifier* [13] đề xuất lưu trữ giá trị được coi là đặc trưng đường đi (PI) của mỗi gói tin khi đi từ nguồn đến đích ứng với mỗi địa chỉ nguồn. Các gói tin đi cùng đường đến đích sẽ có thông tin PI thường giống nhau, và qua đó sẽ được sử dụng để lọc bỏ tất cả các gói tin giả mạo có cùng đường đi với một gói tin giả mạo đã phát hiện trước đó.

Một tiếp cận lý thú khác dựa trên ý tưởng loại bỏ các gói tin dựa trên mức độ tắc nghẽn, trong đó Packetscore [14], một cơ chế điển hình, đề xuất thiết lập mức độ ưu tiên cho mỗi gói tin dựa vào thuật toán *Detecting-Differentiating-Discarding routers* (3D-R) sử dụng độ đo Bayes (Bayesian-theoretic metric). Từ đó, người ta thiết lập cơ chế lọc bỏ các gói tin có mức ưu tiên thấp khi tấn công DDoS xảy ra (gây tắc nghẽn tại các router kề cận).



Hình 2. IP Header.

III. CƠ SỞ LÝ THUYẾT

1. Packet Identification và tính chất tăng dần

Trường thông tin Identification là một trường thông tin 16 bit trong IP Header. Trường thông tin này được sử dụng để xác định thứ tự các gói tin khi bị phân mảnh (fragment) khi gói tin đó được truyền từ môi trường mạng có MTU cao sang môi trường mạng có MTU thấp hơn.

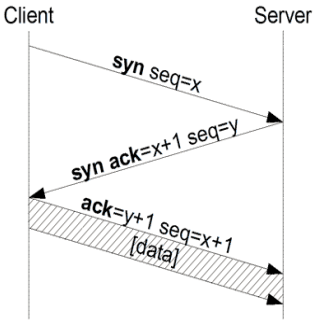
Tuy nhiên, qua quan sát thực tế thì khi hệ thống hoạt động ở trạng thái bình thường, nếu một máy tính gửi ra một gói tin mà không phân biệt địa chỉ IP đích hay dịch vụ sử dụng thì giá trị IPD sẽ tăng lên 1 đơn vị.

Nguyên lý này cũng được áp dụng trong phương pháp Idle scan [15] để kiểm tra một cổng mở trên một máy chủ hay không khi máy tính kiểm tra không thể gửi trực tiếp gói tin đến máy chủ cần kiểm tra mà phải thông qua một máy tính khác ở trạng thái Idle.

Để có thêm sở cứ cho tính chất chúng tôi phát hiện ở trên, chúng tôi kiểm thử lại tính chất trên đối với tập dữ liệu chuẩn DARPA Intrusion Detection Evaluation [16] do viện MIT cung cấp. Tập dữ liệu này được sử dụng nhiều bởi các nghiên cứu [17–19] để đánh giá hiệu quả của các phương pháp phát hiện xâm nhập và tấn công DDoS. Kết quả chúng tôi phát hiện: tấn công DDoS từ địa chỉ nguồn 202.77.162.213 đến địa chỉ 172.16.115.20 qua ứng dụng Telnet có giá trị PID tăng liên tục từ giá trị 11484.

2. Thủ tục bắt tay TCP handshake và tấn công tràn ngập TCP SYN Flood

TCP (Transmission Control Protocol) [20] là giao thức được sử dụng phổ biến trên Internet để tạo một kết nối truyền tin tin cậy giữa 2 máy tính. Theo nguyên lý hoạt động của giao thức này, để khởi tạo một kết nối TCP phải thực hiện quá trình TCP handshakes. Trong quá trình này, đầu tiên, máy khách (client) gửi một gói tin SYN để máy chủ (server) yêu cầu khởi tạo kết nối. Sau khi nhận được yêu cầu, máy chủ sẽ gửi lại gói tin phản hồi SYN-ACK, để thông báo trạng thái sẵn sàng kết nối cho máy khách, đồng thời trong phản hồi đó, máy chủ cũng gửi lại gói tin SYN



Hình 3. TCP handshakes.

để yêu cầu khởi tạo kết nối với phía máy khách. Bước cuối cùng trong quá trình khởi tạo kết nối là máy khách gửi lại gói tin ACK để phản hồi lại cho máy chủ trạng thái sẵn sàng ở phía mình.

Theo cơ chế TCP retransmission [20] của nguyên lý hoạt động của giao thức TCP, trong khoảng thời gian nhất định (time out), sau khi máy khách gửi gói tin SYN để yêu cầu khởi tạo kết nối tới máy chủ, mà nó không nhận được gói tin xác nhận ACK từ phía máy chủ, thì máy khách sẽ gửi lại gói tin SYN khác để tiếp tục yêu cầu kết nối.

Từ nguyên lý khởi tạo kết nối của giao thức TCP ở trên, chúng ta có thể thấy được một điểm yếu rõ ràng: cứ khi nhận được một gói tin SYN, nếu còn đủ tài nguyên thì máy chủ dành sẵn phần tài nguyên bộ nhớ cần thiết để chuẩn bị phục vụ cho kết nối sau này mà không hề kiểm tra tính trung thực của gói tin SYN gửi đến. Lợi dụng điểm yếu này, tin tặc sẽ tấn công máy chủ bằng cách gửi tràn ngập gói tin SYN giả mạo tới máy chủ để làm cạn kiệt tài nguyên trên máy chủ với mục đích là làm máy chủ không còn tài nguyên để phục vụ các yêu cầu hợp lệ gửi tới.

Các gói tin giả mạo được tạo ra với các trường thông tin giống như của gói tin thông thường, nhưng với địa chỉ IP nguồn giả mạo (sinh ngẫu nhiên). Nếu không có cơ chế theo dõi đặc biệt, máy chủ không thể phân biệt được gói tin nào là gói tin thực hay giả mạo.

IV. CƠ CHẾ THUẬT TOÁN LỌC BỎ NHANH GÓI TIN GIẢ MẠO

Trong mục này, chúng tôi sẽ trình bày chi tiết về đề xuất mới của chúng tôi, cơ chế lọc bỏ gói tin tấn công dựa trên việc phát hiện các chuỗi gói tin PID tăng liên tục (nhưng lại có địa chỉ nguồn ngẫu nhiên, không trùng nhau). Đây là những gói tin được coi là giả mạo theo phân tích cơ sở đã nêu trước (mục I-3).

Ở đây, chúng tôi sử dụng các bảng dữ liệu khác nhau để lưu trữ một số thông tin cơ bản của các gói tin gửi đến, như địa chỉ IP và giá trị PID. Các bảng này được tổ chức sao cho việc tìm kiếm hoặc đối chiếu có thể thực hiện rất nhanh (thời gian tìm kiếm có thể coi như hằng số). Từ đó quan hệ đặc biệt giữa các gói tin, như chuỗi PID tăng dần, có thể phát hiện được thông qua tìm kiếm, đối chiếu.

Trong mục này, chúng tôi tập trung mô tả về logic hoạt động của cơ chế thuật toán, chứ không đi sâu mô tả cách thức triển khai cài đặt cụ thể các bảng dữ liệu này (sẽ được đề cập ở mục sau).

Trước hết chúng tôi trình bày cơ chế chung của giải pháp đề xuất tại mục IV-1. Sau đó, trong mục IV-2 chúng tôi trình bày thuật toán PID-Filter cho phép phát hiện dãy PID tăng nhằm loại bỏ các gói tin tấn công. Một số nhận xét phân tích sơ bộ (về mặt lý thuyết) được nêu tại mục IV-3.

1. Cơ chế giải pháp chung

Chúng tôi đề xuất giải pháp chống tấn công SYN Flood để bảo vệ máy chủ bị tấn công bằng cách thực hiện đồng thời hai việc:

- Phát hiện các địa chỉ IP nguồn tốt, đang thực sự có nhu cầu kết nối, hay đã khởi tạo kết nối;
- Phát hiện các gói tin giả mạo thông qua bộ lọc PID-Filter.

Để thực hiện ý đồ cơ bản này, chúng tôi sử dụng hai bảng dữ liệu lưu trữ địa chỉ IP nguồn (có thể kèm thông tin bổ sung):

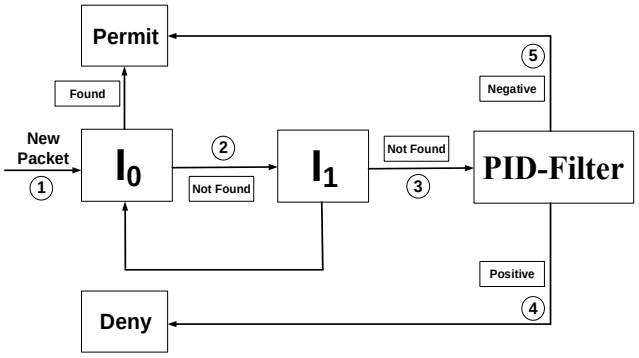
- Bảng I_0 để lưu trữ các địa chỉ IP nguồn sạch và tích cực (đã kết nối, hoặc đang có nhu cầu khởi tạo kết nối thực sự), được sử dụng để cho phép nhanh các máy client đã xác định là hợp lệ kết nối đến hệ thống.
- Bảng I_1 để lưu trữ các thông tin địa chỉ IP nguồn, giá trị PID và thời gian đến của những gói tin mà chưa xác định được là gói tin bình thường (tốt) hay gói tin giả mạo (tấn công). Trong quá trình kiểm tra sau đó, khi một địa chỉ IP trong I_1 được xác định là sạch (âm tính), nó sẽ được chuyển sang I_0.

Sơ đồ hoạt động cơ bản của giải pháp là như Hình 4.

Có thể có những IP giả mạo (tạo ngẫu nhiên) trùng khớp với một địa chỉ IP sạch trong I_0, nhưng xác suất trùng khớp này là nhỏ. Để giảm thiểu ảnh hưởng của vấn đề trùng khớp này, mỗi IP trong bảng I_0 sẽ có ngưỡng thời gian sống nhất định, nếu sau ngưỡng đó mà hệ thống không nhận được kết nối thực từ IP đó thì IP này sẽ bị xóa khỏi bảng I_0.

Một gói tin được xác định là hợp lệ trong các trường hợp sau (và do đó sẽ được chuyển từ I_1 sang I_0):

- Trường hợp có gói tin có cùng địa chỉ IP nguồn đã gửi đến hệ thống trước đó. Trường hợp này xảy ra khi một client đã yêu cầu kết nối tới máy chủ nhưng không được hồi đáp nên gửi lại yêu cầu kết nối, (cơ chế TCP retransmission [20]). Hệ thống có thể phát hiện sai gói



Hình 4. Sơ đồ phương pháp đề xuất.

tin giả mạo là gói tin hợp lệ trong trường hợp địa chỉ giả mạo của gói tin ngẫu nhiên trùng với một địa chỉ nào đó của gói tin đã gửi đến hệ thống trước đó. Tuy nhiên xác xuất trùng khớp này chỉ là $1/2^{32}$ cho mỗi nguồn gửi gói tin giả mạo.

- Trường hợp lệ nếu sau khoảng thời gian δ_T (tham số hệ thống chọn trước) một cặp (IP, PID) trong I_1 được xác định là âm tính qua xét nghiệm PID-Filter.

Mỗi gói tin mới gửi đến hệ thống được xử lý theo các bước sau đây.

Bước 1: Kiểm tra địa chỉ IP nguồn của gói tin có trong bảng I_0 hay không? Nếu có thì cho gói tin vào hệ thống (chuyển tiếp vào máy chủ); Nếu không sang bước 2.

Bước 2: Kiểm tra IP nguồn có trong bảng I_1 hay không? Nếu có thì đưa thông tin IP này vào bảng I_0 (cơ chế TCP retransmission) và cho phép gói tin đi vào hệ thống; Nếu không sang bước các sau.

Bước 3: Lưu thông tin (địa chỉ IP, giá trị PID, thời gian đến) của gói tin hiện thời vào bảng I_1. Triệu gọi PID-Filter với giá trị PID này.

Bước 4: Trong quá trình thực hiện PID-Filter, nếu có các giá trị PID bị phát hiện dương tính thì các thông tin của các gói tin tương ứng sẽ bị loại bỏ khỏi bảng I_1 (song song với việc các gói tin đó bị lọc bỏ, không vào tới máy chủ bên trong).

Bước 5: Nếu PID-Filter báo âm tính, các thông tin tương ứng với giá trị PID này sẽ chuyển khỏi bảng I_1 và đưa vào bảng I_0.

2. PID-Filter: Thuật toán phát hiện dãy PID tăng liên tục

Mục đích của thuật toán là phát hiện ra dãy gói tin có PID tăng dần $x_0 = a$, $x_1 = a + 1$, $x_2 = a + 2$, v.v., trong một khoảng thời gian (rất ngắn) δ_T cho trước. Thuật toán sử dụng hai bảng dữ liệu B_0 và B_1. Bảng B_0 để lưu PID từ một gói tin mới đến, được coi là có tiềm năng trở thành điểm đầu x_0 của một dãy tăng như trên, còn B_1 để lưu giá trị thứ hai x_1 (hoặc điểm cuối hiện thời) của một dãy tăng khi đã phát hiện được. Các giá trị PID từ mỗi gói tin mới đến sẽ được lọc qua PID-Filter. Nếu một giá trị được đưa vào B_0 và trong suốt khoảng thời gian δ_T không bị phát hiện thuộc vào một dãy PID tăng thì PID-Filer sẽ trả kết quả "âm tính" (gói tin tương ứng sẽ được chấp nhận), ngược lại sẽ bị báo "dương tính". Các gói tin tương ứng có giá trị PID trong dãy tăng sẽ bị loại bỏ.

Các bước thực hiện của thuật toán như sau:

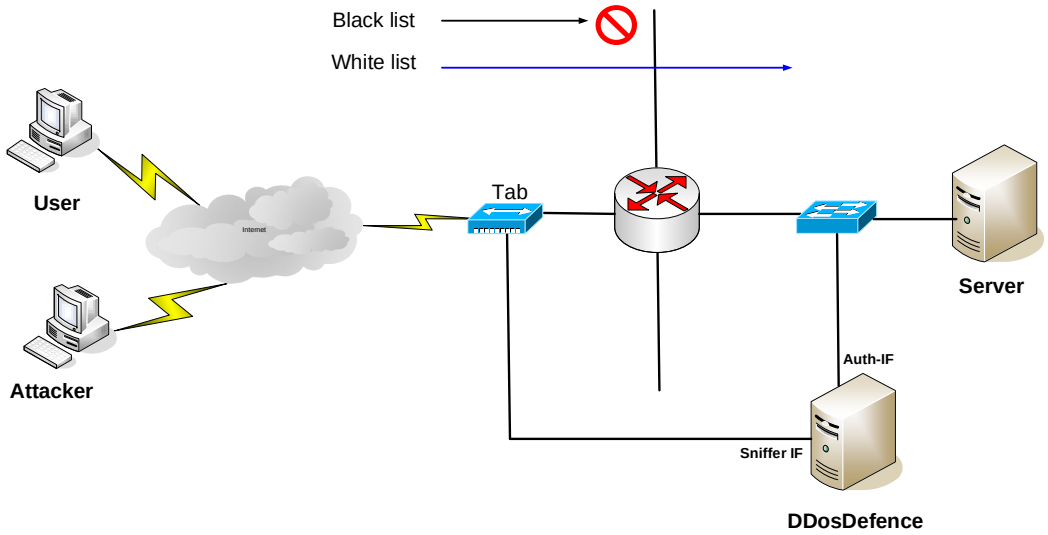
1) Với một gói tin SYN vừa đến, ta lấy ra giá trị x là PID của nó và thực hiện tìm kiếm $x_0 = x - 1$ trên bảng B_0. Nếu thấy, x được coi là giá trị thứ hai của một dãy tăng và được lưu vào B_1 để phát hiện các gói tin có PID tiếp theo trong dãy tăng. PID-Filter dừng (chưa báo kết quả dương tính với giá trị PID x_0 và x mà chờ tiếp).
2) Tìm kiếm $x_1 = x - 1$ trên bảng B_1. Nếu thấy, x được coi là giá trị thứ 3 (hoặc muộn hơn nữa) của một dãy tăng. Giá trị x_1 là đứng ngay trước trong dãy tăng. Giá trị x sẽ được cập nhật thay thế x_1 trong B_1 để tiếp tục đợi các gói tin có PID tiếp theo (thứ 4, 5, v.v.) trong dãy tăng. PID-Filter dừng và trả lại kết quả dương tính của cả giá trị PID x và x_0.
3) Tìm kiếm $x_0 = x - 2$ trên bảng B_0. Nếu thấy, ta sẽ lưu x vào B_1 và xử lý tương tự như ở bước 2. Sở dĩ làm như vậy là vì ta coi như có thể gói tin có PID là $x_1 = x - 1$ đã bị thất lạc hoặc đến muộn hơn sau này. PID-Filter dừng và trả lại kết quả dương tính với giá trị PID x và x_0.
4) Các bước tìm kiếm ở trên đều có kết quả là không thấy: Ta có thể coi x có tiềm năng là điểm đầu x_0 của một dãy tăng mới và cập nhật nó vào B_0.

Bên cạnh các bước "xét nghiệm" như trên, một hoạt động then chốt khác của thuật toán là tiếp tục kiểm tra bảng I_1 (có thể chạy ngầm) để phát hiện các giá trị PID và IP tương ứng đã đủ tuổi (δ_T) ra khỏi bảng mà không hề bị "liên đới" với các các phát hiện dương tính có thể có ở bước 1 hoặc 3. Điều đó có nghĩa là giá trị x như vậy thực ra là của một gói tin lành, không còn bị nghi ngờ nữa. Các gói tin tương ứng sẽ được báo âm tính.

Để đảm bảo xác suất lỗi thấp (lọc bỏ nhầm) bảng B_1 cũng cần được làm tươi (refresh) liên tục. Các giá trị PID đã cũ (có "tuổi" vượt ngưỡng δ_T) sẽ bị loại bỏ ngay khỏi bảng. Thuật toán của phương pháp được mô tả trong Thuật toán 1.

3. Mô hình hệ thống

Phương pháp của chúng tôi không can thiệp, xử lý trực tiếp các kết nối mạng gửi đến máy chủ. Nó cho phép xây dựng một danh sách địa chỉ IP sạch (White-List).



Hình 5. Mô hình hệ thống triển khai xử lý tấn công DDoS.

Thuật toán 1: Thuật toán phương pháp đề xuất

for mỗi gói tin mới $P(i)$
  If (SrcIP[$P(i)$] thuộc I_0)
    Cho phép $P(i)$;
  else if (SrcIP[$P(i)$] thuộc I_1)
    Thêm SrcIP[$P(i)$] vào I_0
    Loại SrcIP[$P(i)$] khỏi I_1)
  //*Thực hiện bước 1 trong PID-Filter*
  else if (PID($x-1$)[$P(i)$] thuộc B_0)
    Thêm PID(x)[$P(i)$] vào B_1
  //*Thực hiện bước 2 trong PID-Filter*
  else if (PID($x-1$)[$P(i)$] thuộc B_1)
    Cập nhật B_1[PID($x-1$)[$P(i)$]] = B_1[PID(x)[$P(i)$]]
    Cảnh báo PID($x-1$)[$P(i)$] và PID($x-2$)[$P(i)$]
  //*Thực hiện bước 3 trong PID-Filter*
  else if (PID($x-2$)[$P(i)$] thuộc B_0)
    Cập nhật B_1[PID($x-1$)[P(i)]] = B_1[PID(x)[$P(i)$]]
    Cảnh báo PID($x-1$)[$P(i)$] and PID($x-2$)[$P(i)$]
  //*Thực hiện bước 4 trong PID-Filter*
  else
    Thêm PID(x)[$P(i)$] vào B_0
  end
end

Khi tấn công DDoS xảy ra danh sách địa chỉ IP trong White-List sẽ được gửi đến máy chủ hoặc thiết bị mạng để thực hiện chính sách khóa các địa chỉ IP không nằm trong White-List này. Danh sách địa chỉ IP trong White-List cũng liên tục cập nhật thông qua thuật toán của chúng tôi. Cụ thể ở đây là các địa chỉ IP trong I_0 sẽ được đưa vào White-List. Trong mô hình trên, phương pháp của chúng tôi được cài đặt trên máy chủ DDoS Defence. Máy chủ này có hai giao diện. Một giao diện được sử dụng để thu thập thông tin trên giao diện kết nối Internet của Router biên của hệ thống (Sniffer IF) sử dụng thiết bị trích rút dữ liệu chuyên dụng (Network-Tap) nhằm không làm ảnh hưởng tới hoạt động của hệ thống được bảo vệ. Giao diện thứ hai được sử dụng để gửi danh sách IP trong White-list đến máy chủ hoặc thiết bị mạng.

4. Phân tích thuật toán

Sau đây, chúng tôi đưa ra một phân tích đánh giá về giải pháp này trên góc độ lý thuyết. Ta hãy phân tích về hoạt động của thuật toán PID-Filter. Giả sử một máy tính (máy bị điều khiển để tấn công DDoS - máy bot) tạo và gửi một chuỗi gói tin SYN tấn công ký hiệu là $P_0, P_1, P_2, P_3, \ldots$, với dãy giá trị PID tăng dần. Khoảng thời gian δ_T được đặt đủ dài hơn khoảng thời gian giữa hai gói tin P_0 và P_3 đến đích (máy chủ bị tấn công). Nếu như đường truyền tốt và không có gì đặc biệt xảy ra thì chuỗi gói tin sẽ đến đúng theo thứ tự ban đầu, chúng sẽ lần lượt được cập nhật địa chỉ IP vào bảng I_1, giá trị PID của P_0 vào bảng B_0 (chỉ mình P_0 tại bước 4), và vào B_1 (P_1, P_2, trở đi tại bước 2). Tuy nhiên, khi nhận được P_2, chuỗi gói tin tấn công bị nhận diện nên các địa chỉ IP tương ứng bị loại bỏ khỏi bảng I_1, đồng thời PID của P_0 được loại bỏ trên B_0, các PID của các gói P_1 và tiếp theo cũng bị loại bỏ trên B_1, ngoại trừ của gói tin cuối cùng (P_i) được lưu ở đây để tiếp tục phát hiện những gói tin tấn công tiếp theo sau này cùng một chuỗi đó ($P_{i+1}, \ldots$).

Chúng ta xét tiếp trường hợp có những trục trặc (do kết nối mạng đang có tắc nghẽn hoặc sự cố trên đường truyền),

trật tự gói tin đến có thể thay đổi hoặc có gói tin thất lạc. Nếu máy chủ nhận được dãy các gói tin P_0, P_1, P_3, hay P_0, P_2, P_3 (tức là chỉ 1 gói bị thất lạc) thì chuỗi tấn công vẫn có thể bị phát hiện, tuy nhiên nếu có 2 gói tin thất lạc trở lên trong số 4 gói đầu thì sẽ không phát hiện được. Đây có thể coi là điểm yếu còn tồn tại của thuật toán, cần được cải tiến nâng cao trong tương lai.

Khi mỗi gói tin lẻ không phát hiện thuộc dãy tăng nào (hoặc là bị sót như trường hợp cặp P_0, P_1, rồi P_2, P_3 bị thất lạc hoặc đến rất muộn) có thời gian tồn tại vượt δ_T thì các thông tin tương ứng sẽ bị loại khỏi B_0, B_1 và I_1 nhưng địa chỉ IP sẽ được cập nhật vào I_0 - danh sách các IP được coi là sạch trong hiện tại[3].

Sau đây chúng tôi phân tích về các lỗi nhầm lẫn có thể xảy ra do ngẫu nhiên. Xác suất một gói tin giả mạo có thể may mắn được chấp nhận (tức là báo âm tính nhầm - False Negative) vì có địa chỉ IP tình cờ trùng khớp với địa chỉ nào đó trong I_0 là $p_0 = |I_0| / 2^{32}$.

Xác suất một gói tin giả mạo có địa chỉ IP được đưa vào I_0 vì tình cờ trùng khớp địa chỉ nào đó trong I_1 (tức là được coi là tốt theo cơ chế TCP retransmission - False Nagative) là $p_1 = |I_1| / 2^{32}$.

Xác suất một gói tin tốt có giá trị PID tình cờ lớn hơn 1 so với một giá trị trong B_0 (tức là báo dương tính nhầm - False Positive) là $p_2 = |B_0| / 2^{16}$.

Các xác suất nói trên có thể được ước lượng như sau. Giả sử tại một thời điểm đang diễn ra tấn công, có N máy tấn công và M máy khách hàng tốt đang tìm cách kết nối đến máy chủ. Nếu giả định rằng số lượng các trục trặc về truyền tin trên mạng (gói tin thất lạc hoặc đi vòng lâu hơn nhiều) là đủ nhỏ thì ta dễ thấy: $|I_0| \approx M$, $|I_1| \leq M + 3N$, $|B_0| \approx N$ và $|B_1| \leq 2N$. Các ước lượng về $|I_1|$, $|B_0|$ và $|B_1|$ có được là do theo phân tích ở trên về quá trình hoạt động của PID-Filter dễ thấy tại mỗi thời điểm chỉ có từ 2 đến 3 gói tin của cùng một chuỗi tăng (phát đi từ một máy bot) được ghi nhận trong I_1.

Bảng I minh họa cho các phân tích và ước lượng trên. Con số máy tham gia tấn công được giả định từ cấp độ qui mô trung bình (từ 1.000 đến 2.000) đến rất cao (lớn hơn 10.000), dựa trên các báo cáo về qui mô các tấn công SYN thường thấy. Có thể thấy rằng các xác suất lỗi bỏ sót (âm tính sai) đều rất nhỏ, trong khi xác suất lỗi báo nhầm tấn công (dương tính sai) có thể lớn đáng kể khi qui mô số máy tấn công là lớn cỡ 10.000 trở lên.

V. ĐÁNH GIÁ THỰC NGHIỆM BAN ĐẦU

Trong triển khai cài đặt, hiện nay chúng tôi lựa chọn giá trị $\delta_T \approx$ 1–2 ms, cho phép có thể theo dõi được chuỗi gói tin tăng PID phát từ một máy bot có băng thông tấn công vào cỡ chỉ 1 Mb, tức là được coi vào loại nhỏ nhất đáng kể (nếu băng thông tấn công nhỏ hơn nữa ta có thể coi là quá bé nên không cần quan tâm nữa). Các bảng lưu trữ được cài đặt bằng cấu trúc dữ liệu mảng bản ghi đơn giản.

[3]Chỉ là sạch trong phạm vi cơ chế đề xuất của chúng tôi. Bên cạnh đó, các vành đai kiểm soát khác của Firewall vẫn có thể kiểm soát nhưng danh sách IP sạch khác, hoạt động độc lập.

Bảng I
QUI MÔ TẤN CÔNG GIẢ ĐỊNH VÀ XÁC SUẤT LỖI

Giá trị giả định				**Xác suất nhận được**		
M	N	$\|I_1\|$	$\|B_1\|$	P_0	P_1	P_2
1.000	1.000	4.000	2.000	2, 3E-07	0, 93E-06	1,5%
1.500	2.000	7.500	4.000	3, 5E-07	1, 74E-06	3%
2.000	4.000	14.000	8.000	4, 7E-07	3, 25E-06	6%
2.500	8.000	26.500	16.000	5, 8E-07	6, 17E-06	12%
3.000	16.000	51.000	32.000	7, 0E-07	11, 8E-06	24%

Theo những nghiên cứu mà chúng tôi đề cập ở mục II, thì những nghiên cứu này không sử dụng một dữ liệu mẫu chung nào để đánh giá thực nghiệm. Tập dữ liệu kiểm thử DARPA của Viện MIT có dữ liệu kiểm thử cho một số dạng tấn công Dos/DDoS, tuy nhiên tập dữ liệu này lại không có dữ liệu kiểm thử cho dạng tấn công SYN Flood mặc dù có đề cập tới [16].

Do đó, trong nghiên cứu này, chúng tôi sử dụng dữ liệu thử nghiệm do chúng tôi tự thu thập được từ một cuộc tấn công SYN Flood trên môi trường thực với 336.671 gói tin được ghi nhận (thời điểm chúng tôi lấy mẫu hệ thống đã quá tải và dừng hoạt động tại con số nói trên). Theo đánh giá chuyên gia, có thể coi toàn bộ gói tin mà chúng tôi thu thập được là gói tin giả mạo, số lượng gói tin thực nếu có thì chiếm tỷ lệ rất nhỏ, không đáng kể. Mỗi gói tin thu được, chúng tôi thu thập những thông tin: thời gian đến của gói tin (timestamp), địa chỉ IP nguồn (Src IP), cờ trạng thái kết nối (flag) và giá trị PID.

Kết quả thực nghiệm được đưa ra trong Bảng II, bao gồm các giá trị sau: Số lượng gói tin trong mẫu thử nghiệm (#Packets), số lượng gói tin giả mạo phát hiện được (#Detected Packets), số lượng các chuỗi có PID tăng dần được tạo ra (#PID-Groups) và tỷ lệ (Rate) các gói tin giả mạo phát hiện. Các kết quả đều cho thấy tỉ lệ phát hiện được là rất cao.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi đã đề xuất một cơ chế thuật toán mới để phát hiện và lọc bỏ những gói tin giả mạo sử dụng trong tấn công DDoS dạng SYN Flood mà không phải thực hiện các bước chuẩn bị gián tiếp tốn kém như quá trình học mẫu thường được sử dụng phổ biến (chúng tôi cũng đã sử dụng trước đây [3]).

Bảng II
KẾT QUẢ THỰC NGHIỆM

#Packets	#Detected Packets	#PIDGroups	Rate
100.000	92.954	17.043	92,95%
150.000	139.189	25.479	92,79%
200.000	185.634	33.980	92,81%
250.000	232.061	42.574	92,82%
300.000	278.444	50.959	92,81%
336.671	312.583	57.179	92,84%

Các đánh giá trên các phương diện phân tích lý thuyết cũng như thực nghiệm cho thấy giải pháp của chúng tôi là rất khả quan khi qui mô số máy tấn công là không quá lớn (nhỏ hơn 10.000 máy). Phương pháp xử lý các gói tin cho phép đạt tốc độ rất nhanh, đáp ứng đòi hỏi đặc biệt của việc chống tấn công DDoS.

Khi số máy tấn công là rất lớn, tỷ lệ báo dương tính nhầm có thể tăng cao (trên 10% trở lên), làm ảnh hưởng đáng kể tới việc kết nối của các máy khách hàng tốt. Cơ chế kiểm soát của chúng tôi cũng dựa trên giả định là tỷ suất truyền tin trục trặc hay thất lạc (phía bên ngoài máy chủ) là đủ thấp.

Một tồn tại khác là chúng tôi chưa đề xuất được phương án đặc thù cho lưu trữ và tìm kiếm nhanh các thông tin trên các bảng I_0, I_1, B_0 và B_1. Khi qui mô tấn công rất lớn, không gian lưu trữ mỗi bảng và thời gian tìm kiếm thông tin trên mỗi bảng sẽ lớn, thời gian xử lý có thể trở nên chậm đáng kể.

Trong các nghiên cứu tiếp theo, chúng tôi sẽ tiếp tục đề xuất thuật toán nâng cao để khắc phục các lỗi bỏ sót đã phân tích (mục IV-3) cũng như xây dựng các phương pháp đặc thù, hiệu quả hơn cho lưu trữ và xử lý thông tin trên các bảng I_0, I_1, B_0 và B_1 để tối ưu hơn nữa các yếu tố hiệu năng (thu gọn không gian lưu trữ và giảm tốc độ xử lý thông tin trên mỗi bảng).

LỜI CẢM ƠN

Chúng tôi xin cảm ơn sự hỗ trợ của Cục An toàn Thông tin của Bộ Thông tin và Truyền thông, thông qua Nhiệm vụ Khoa học Công nghệ cấp Quốc gia: "Nghiên cứu, xây dựng Khung tham chiếu về an toàn thông tin phục vụ chính phủ điện tử", mã số KC.01.07/16-20.

TÀI LIỆU THAM KHẢO

[1] C. C. Center, "TCP SYN flooding and IP spoofing attacks," *CERT Advisory CA-96-21*, Sep. 1996.

[2] Internet Engineering Task Force, "Internet protocol," Standard RFC 791, 1981.

[3] T. M. Thang and V. K. Nguyen, "Synflood Spoof Source DDoS Attack Defence Based on Packet ID Anomaly Detection-PIDAD," in *Information Science and Applications (ICISA)*. Springer, 2016, pp. 739–751.

[4] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[5] H. Zimmermann, "OSI reference model–The ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.

[6] A. John and T. Sivakumar, "DDoS: Survey of traceback methods," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 241–245, 2009.

[7] A. C. Snoeren, "Hash-based IP traceback," in *In Proceedings of the ACM SIGCOMM*. Citeseer, Aug. 2001, pp. 3–14.

[8] J. B. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, "Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study," in *Proceedings of the IEEE/IFIP International Symposium on Integrated Network Management*. IEEE, 2001, pp. 609–622.

[9] R. Jalili, F. Imani-Mehr, M. Amini, and H. Shahriari, "Detection of distributed denial of service attacks using statistical pre-processor and unsupervised neural networks," *Information Security Practice and Experience*, pp. 192–203, 2005.

[10] M. Li, J. Liu, and D. Long, "Probability Principle of a Reliable Approach to Detect Signs of DDOS Flood Attacks." in *Proceedings of the Parallel and Distributed Computing: Applications and Technologies*. Springer, 2004, pp. 596–599.

[11] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based IP filtering," in *Proceedings of the IEEE International Conference on Communications (ICC'03)*, vol. 1. IEEE, 2003, pp. 482–486.

[12] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Transactions on Networking (ToN)*, vol. 15, no. 1, pp. 40–53, 2007.

[13] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in *Proceedings of the Symposium on Security and Privacy*. IEEE, 2003, pp. 93–107.

[14] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 2, pp. 141–155, 2006.

[15] F. Almeida, "idlescan (ip.id portscanner)," 1999.

[16] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[17] U. Akyazi and A. S. Uyar, "Distributed detection of DDoS attacks during the intermediate phase through mobile agents," *Computing and Informatics*, vol. 31, no. 4, pp. 759–778, 2012.

[18] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[19] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International Journal of Computer Science and Network Security*, vol. 7, no. 12, pp. 258–263, 2007.

[20] Internet Engineering Task Force, "Transmission control protocol," Standard RFC 793, 1981.

Trần Mạnh Thắng sinh năm 1982, làm việc tại Cục An toàn thông tin - Bộ Thông tin và Truyền thông. Hiện nay, ông là nghiên cứu sinh năm thứ ba tại Bộ môn Công nghệ Phần mềm, Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách khoa Hà Nội. Hướng nghiên cứu của ông là phòng, chống tấn công từ chối dịch vụ.

Nguyễn Khanh Văn sinh năm 1970 tại Hà Nội. Ông tốt nghiệp Trường Đại học Bách khoa Hà Nội năm 1992. Ông nhận bằng Thạc sĩ, năm 2000, tại Trường Đại học Wollogong, Úc và nhận bằng Tiến sĩ, năm 2006, tại Trường Đại học California - Davis, Hoa kỳ. Hiện nay, ông đang công tác tại Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách khoa Hà Nội. Hướng nghiên cứu chính của ông là tính toán phân tán, mô hình mạng mới và thuật toán, mạng máy tính và an toàn thông tin.