NGHIÊN CỨU GIẢI PHÁP TẤN CÔNG MẠNG ẢO HOÁ SỬ DỤNG ACK-STORM VÀ PHƯƠNG PHÁP PHÒNG CHỐNG

Trần Nam Khánh*, **Nguyễn Kim Thanh***, **Phùng Vũ Hạnh My**, **Tạ Minh Thanh**‡*
*Học Viện Kỹ Thuật Quân Sự, 236 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội

Tóm tắt*—Trong những năm gần đây, thị trường máy chủ ảo hóa (hypervisor) đang phát triển nhanh chóng bởi công nghệ máy chủ ảo hóa đem lại nhiều lợi ích cho các dịch vụ thực tế. Ngày càng có nhiều doanh nghiệp sử dụng máy chủ ảo hóa như một giải pháp thay thế cho các máy chủ vật lý. Tuy nhiên, các nghiên cứu an ninh mạng cho thấy các máy chủ ảo hóa có nhiều lỗ hổng an toàn mạng hơn các máy chủ truyền thống. Chính vì điều đó, đứng trên vai trò của một nhà quản trị hệ thống, việc chuẩn bị cho các tình huống xấu nhất, hiểu rõ các mối nguy hại và nghiên cứu thêm các mối nguy hiểm tiềm tàng có thể làm tổn hại đến máy chủ ảo, các hệ thống ảo là cần thiết. Trong bài báo này, chúng tôi tập trung vào khai thác các lỗ hổng bảo mật chưa được chú ý nhiều trên các hệ thống ảo hoá và thực hiện mô phỏng tấn công từ chối dịch vụ (DoS) sử dụng bão gói tin ACK (ACK-Storm) bằng các gói tin giả mạo trên mạng ảo VMWare và mạng Docker. Từ đó đề xuất một số biện pháp phòng chống những kỹ thuật tấn công đã thử nghiệm trong bài báo

***Từ khóa*—Tấn công từ chối dịch vụ-DoS, Hypervisor, Giao thức TCP, ACK-Storm, Mạng ảo, Mạng Docker.**

I. GIỚI THIỆU

Những báo cáo gần đây chỉ ra rằng máy chủ ảo hóa đang ngày càng trở nên phổ biến và được sử dụng một cách rộng rãi hơn. Vào tháng 11 năm 2012, Jake McTigue đã đề cập trong "Khảo sát quản lý ảo hóa năm 2013" [7] của "Information Week", 42% trong số 320 doanh nghiệp công nghệ được khảo sát cho biết công ty của họ sử dụng nhiều máy chủ ảo hóa và mức sử dụng sẽ đạt tới 87% vào năm 2013. Sau đó, vào tháng 9 năm 2019, Spiceworks tiến hành khảo sát [1] trên 539 nhà đưa ra quyết định công nghệ thông tin của các doanh nghiệp với đa dạng quy mô, bao gồm các doanh nghiệp nhỏ, doanh nghiệp vừa và các tập đoàn lớn, và trên nhiều lĩnh vực, bao gồm giáo dục, chăm sóc sức khỏe, tổ chức phi lợi nhuận, chính phủ, tài chính, bán lẻ, xây dựng, sản xuất và dịch vụ công nghệ thông tin. Cuộc khảo sát của Spiceworks chỉ ra rằng hầu hết các doanh nghiệp đều sử dụng công nghệ ảo hóa máy chủ, với 92% đang sử dụng và 5% dự định sử dụng trong vòng 2 năm tiếp theo (tính từ năm 2019). Đối với các doanh nghiệp, điều quan trọng là phải đạt được hiệu quả tối đa với chi phí tối thiểu, và chi phí tối ưu cho hạ tầng công nghệ thông tin là một trong những lợi ích mà công nghệ ảo hóa đem lại. Công nghệ ảo hóa, đặc biệt là ảo hóa máy chủ, đang hoạt động rất tốt. Để cung cấp dịch vụ lưu trữ web và sử dụng hiệu quả các nguồn lực hiện có trong cơ sở hạ tầng công nghệ thông tin theo cách tiết kiệm chi phí hơn, chúng ta có thể sử dụng các máy chủ ảo thay cho các máy chủ vật lý. Trước khi có máy chủ ảo, các máy chủ vật lý chỉ sử dụng một phần công suất xử lý của chúng và hoạt động kém hiệu quả hơn vì chỉ một phần máy chủ của mạng xử lý khối lượng công việc. Điều này dẫn đến việc các trung tâm dữ liệu trở nên quá tải với các máy chủ hoạt động kém hiệu quả, gây lãng phí tài nguyên và điện năng. Để khắc phục tình trạng này, ảo hóa máy chủ chia mỗi máy chủ vật lý thành nhiều máy chủ ảo và cho phép mỗi máy chủ hoạt động như một thiết bị vật lý độc lập với khả năng chạy các ứng dụng và hệ điều hành của riêng nó. Bằng cách này, ảo hóa máy chủ mang lại nhiều lợi ích cho người sử

Tác giả liên hệ: Tạ Minh Thanh,
Email: thanhtm@mmta.edu.vn
Đến tòa soạn: 4/2021, chỉnh sửa: 5/2021, chấp nhận đăng: 5/2021.
‡ Corresponding author

dụng: hiệu năng của máy chủ cao hơn, giảm thiểu chi phí vận hành hệ thống và độ phức tạp của máy chủ, tăng hiệu suất ứng dụng và đẩy nhanh tốc độ triển khai công việc.

A. Thách thức của ảo hóa máy chủ

Trong khi mức độ phổ dụng của công nghệ ảo hóa máy chủ ngày càng tăng cao, và những ích lợi nó đem lại là vô cùng to lớn, các vấn đề về an toàn mạng cho hệ thống máy chủ ảo càng trở nên quan trọng và mỗi nhà quản trị hệ thống cần quan tâm hơn nữa bởi vì có rất nhiều mối đe doạ an toàn mạng tiềm tàng đối với một hệ thống máy chủ nói chung và với hệ thống máy chủ ảo nói riêng. Một nghiên cứu thống kê các lỗ hổng và cách phòng chống với các loại lỗ hổng bảo mật trên các hệ thống ảo [25] được thực hiện vào năm 2019 chỉ ra rằng các kỹ thuật tấn công được xây dựng chủ yếu dựa trên các lỗ hổng từ phía Hypervisor, trình quản lý máy ảo, lên tới 61%. Điều này cho thấy trình quản lý ảo hóa vẫn có nhiều lỗ hổng bảo mật cần khắc phục để nâng cao chất và lượng hiệu quả sử dụng của công nghệ ảo hóa. Trong quá trình nghiên cứu, chúng tôi nhận thấy kỹ thuật tấn công DoS ACK-Storm cũng dựa trên lỗ hổng bảo mật của trình quản lý ảo hóa và sẽ phân tích cụ thể hơn trong phần trình bày các thực nghiệm. Trong phạm vi nghiên cứu của bài báo, chúng tôi tập trung khắc phục một số vấn đề an ninh mạng sau:

1) Vào năm 2019, Impreva [2] đã công bố về một cuộc tấn công từ chối dịch vụ phân tán sử dụng gói tin SYN vào tháng 1, trong đó 500 triệu gói tin SYN được gửi mỗi giây. Cũng trong năm 2019, một báo cáo khác của Imperva vào tháng 4 [2] công bố một cuộc tấn công từ chối dịch vụ phân tán với 580 triệu gói tin mỗi giây. Với mỗi gói tin có giá trị trung bình vào khoảng 850 bytes, cuộc tấn công với 580 triệu gói tin mỗi giây, tương đương với lưu lượng mạng là 3944 Gbps, gửi tới mục tiêu để khiến nó không thể phản hồi các yêu cầu hợp lệ được gửi đến. Trước đó, vào năm 2018, Github đã duy trì cuộc tấn công từ chối dịch vụ phân tán với lưu lượng mạng lên tới 1.35Tbps (tương đương 129.6 triệu gói tin mỗi giây) mà không cần đến sự giúp đỡ của mạng botnet. Số lượng các cuộc tấn công từ chối dịch vụ phân tán đang gia tăng nhanh chóng qua từng năm, gấp 3 lần chỉ sau 1 năm, đang làm cho các doanh nghiệp thất thoát rất nhiều tiền của. Để giảm thiểu và ngăn chặn các thiệt hại từ các cuộc tấn công từ chối dịch vụ, việc nghiên cứu và phân tích các loại hình tấn công từ chối dịch vụ là rất cần thiết.
2) Nếu những kẻ tấn công kết hợp kỹ thuật tấn công DoS ACK-Storm với một mạng botnet, chúng có thể khiến máy chủ ảo rơi vào tình trạng cạn kiệt cổng trong một khoảng thời gian. Điều đó ảnh hưởng đến các dịch vụ thực sự, khiến các dịch vụ này không thể đáp ứng các yêu cầu từ người dùng thật sự. Trong các ứng dụng thực, người dùng sẽ ngưng sử dụng các dịch vụ của một nhà cung cấp nếu có vấn đề về thời gian phản hồi và hiệu suất. Vì vậy, việc phát hiện các cuộc tấn công mạng trên mạng ảo (thậm chí chỉ xảy ra trong một khoảng thời gian) để ngăn chặn nó là khá quan trọng đối với các ứng dụng, dịch vụ thực tế như các ứng dụng, dịch vụ dựa trên nền tảng web và API.
3) Sau khi phát hiện các cuộc tấn công mạng nhằm vào các hệ thống ảo, làm thế nào để ngăn chặn các cuộc tấn công và tránh rủi ro, cũng là một vấn đề thách thức quan trọng. Do đó, chúng tôi đề xuất nghiên cứu để triển khai được các giải pháp nâng cao tính bảo mật của mạng ảo trong thực tế.

B. Đóng góp của bài báo

Dựa trên các hiểu biết của chúng tôi, những nghiên cứu về các cuộc tấn công nhằm vào vSwitch/vBridge của các hệ thống ảo trên môi trường như VMware, Docker không được tập trung nhiều, đặc biệt là trên Docker. Do đó, các dịch vụ thực được triển khai trên hệ thống ảo trên môi trường như VMware, Docker dễ bị tấn công mạng hơn các hệ thống thật (các hệ thống vật lý). Đây chính là động lực của chúng tôi khi thực hiện nghiên cứu các cuộc tấn công mạng liên quan vào các hệ thống ảo như vậy. Một cách khái quát, chúng tôi trình bày ngắn gọn những đóng góp của chúng tôi trong bài báo này như sau:

1) Đề xuất thử nghiệm tấn công từ chối dịch vụ dựa trên bão gói tin ACK bằng gói tin FIN-ACK giả mạo trên các dịch vụ được

triển khai trên hệ thống ảo để các nhà cung cấp dịch vụ hiểu được rủi ro và biết thêm lỗ hổng bảo mật khi triển khai dịch vụ trong môi trường ảo. Các gói tin FIN-ACK trên hệ thống ảo hoá dễ dàng bị giả mạo và sẽ làm cạn kiệt tài nguyên của máy ảo chạy dịch vụ. Điều đó có nghĩa rằng các ứng dụng, dịch vụ thực được triển khai trên hệ thống ảo có thể dễ dàng bị tin tặc tấn công qua Internet nếu không được quan tâm đúng mức.

2) Đề xuất tấn công từ chối dịch vụ thực hiện trên các hệ thống máy ảo VMWare và Docker trong bài báo này là đề xuất đầu tiên được thực hiện để chứng minh tính khả thi khi tin tặc muốn tấn công các dịch vụ trên hệ thống máy chủ ảo như VMWare, Docker. Chúng tôi đề xuất một phương pháp tấn công mới dựa trên kỹ thuật tấn công DoS ACK-Storm bằng gói FIN-ACK giả có thể khiến vSwitch/vBridge rơi vào tình trạng cạn kiệt cổng trong một khoảng thời gian.
3) Dựa trên các thí nghiệm tấn công này, bài báo của chúng tôi cũng đưa ra một số giải pháp để ngăn chặn và giảm sự phá hủy của các kiểu tấn công này trong các ứng dụng thực tế.

C. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo

Để thực hiện kỹ thuật tấn công DoS ACK-Storm, trước hết, kẻ tấn công cần thực hiện thành công kỹ thuật tấn công xen giữa (Man-in-the-middle). Kỹ thuật tấn công xen giữa là một thuật ngữ chung [26] [28] để chỉ hành vi của kẻ tấn công mạng tự đặt bản thân vào giữa giao tiếp mạng của một kết nối hợp lệ bằng cách giả dạng một trong các bên và duy trì kết nối hoạt động giống như bình thường nhằm đánh cắp, sửa đổi và ngăn chặn giao tiếp giữa các bên của kết nối. Một kẻ tấn công xen giữa mạnh mẽ với đầy đủ khả năng có thể gây ra rất nhiều thiệt hại cho các máy chủ, nhưng trong thực tế việc thực hiện được điều này là khó khăn hơn nhiều so với các mô hình thí nghiệm với các giao thức bảo mật mạnh mẽ như HTTPS, SSH, VPN [28]. Tuy nhiên, một trong các kỹ thuật phòng thủ hiệu quả sử dụng trao đổi khóa Diffie-Hellman đã được chứng minh là hoàn toàn có thể bị đánh bại bởi thuật toán hiện đại như Field Sieve [27] [28] cho thấy không có một hệ thống mạng nào an toàn tuyệt đối. Đồng thời, trong một nghiên cứu về một số kỹ thuật tấn công mạng (trong đó có tấn công xen giữa) áp dụng với môi trường ảo hóa [29] chỉ ra rằng các switch/bridge ảo thiếu các biện pháp kiểm soát bảo mật hơn nhiều so với các thiết bị vật lý tương ứng. Trong khi đó, để thực hiện kỹ thuật tấn công DoS ACK-Storm chỉ cần sử dụng đến mô hình tấn công xen giữa "yếu" [3], có nghĩa là, kẻ tấn công chỉ cần có khả năng nghe trộm gói tin và gửi một số gói tin giả mạo vào kết nối.

Ý tưởng của kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo [3] được đề xuất bởi tác giả Abramov và Herzberg dựa vào lỗ hổng trong xử lý ngoại lệ của TCP được mô tả trong trang 72 của RFC 793 [11] về TCP. Cụ thể, khi kết nối TCP đang ở trạng thái ESTABLISHED (đã kết nối), nhận được một gói tin với giá trị số xác nhận "chưa được gửi" SEG.ACK >SND.NXT (Giá trị số xác nhận nhận được cao hơn số thứ tự phân đoạn của byte dữ liệu tiếp theo được gửi đến thiết bị còn lại của kết nối TCP), bên nhận sẽ xử lý như sau: Gửi một gói tin ACK (với số SEQ/ACK được gửi cuối cùng) cho bên còn lại của kết nối, sau đó bỏ phân đoạn lỗi đi. Đặc biệt, bỏ qua payload trong phân đoạn. Tuy nhiên, trạng thái này có "timeout" và dừng khi bộ định thời đạt đến timeout. Để thực hiện kỹ thuật tấn công DoS ACK-Storm cơ bản (Tấn công bão gói tin ACK sử dụng hai gói tin ACK giả mạo), kẻ tấn công thực hiện như sau:

- Bắt (ít nhất) một gói tin từ kết nối TCP giữa máy khách và máy chủ (Chỉ cần bắt/nghe lén một gói tin và không cần bất kỳ tác động nào đến kết nối).
- Tạo hai gói tin, mỗi gói tin được gửi đến một bên trong kết nối và với địa chỉ người gửi của bên kia ($i.e.$ gói tin giả mạo). Các gói tin giả mạo phải nằm trong cửa sổ TCP của cả hai bên. Các gói tin giả mạo phải có nội dung - ít nhất là một byte dữ liệu (nếu không có dữ liệu thì sẽ không được xử lý).
- Gửi các gói tin đến máy khách và máy chủ cùng một lúc. Kết nối TCP sau đó sẽ rơi vào một vòng lặp vô hạn gửi các gói ack qua lại giữa hai bên cho đến khi bị timeout.

D. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo

Dựa trên ý tưởng của tác giả Abramov [3], Sơn $et\ al.$ đã đề xuất một kỹ thuật tấn công DoS ACK-

Storm khác [5] để tấn công các máy ảo với cùng một cơ chế. Tuy nhiên, kỹ thuật tấn công được thực hiện bởi các gói tin FIN-ACK giả được tạo cùng một cách khi tạo cặp gói tin ACK giả để thực hiện kỹ thuật tấn công DoS ACK-Storm . Dựa trên mô tả trong RFC-793 [11] (trang 73): nếu kết nối TCP ở trạng thái CLOSE-WAIT (đợi đóng kết nối), nó sẽ xử lý tương tự như đang ở trạng thái ESTABLISHED. Điều đó có nghĩa là nếu những kẻ tấn công buộc mỗi bên của kết nối TCP chuyển sang trạng thái CLOSE-WAIT, thì các bên của kết nối đó sẽ đợi gói tin ACK (xem Hình 2) không bao giờ đến mà thay vào đó là các gói tin ACK "chưa được gửi", sau đó, cuộc tấn công từ chối dịch vụ dựa trên bão gói tin ACK được kích hoạt bởi hai gói tin FIN-ACK giả. Vì cấu hình mặc định của giao thức TCP không có "timeout" cho trạng thái CLOSE-WAIT, cuộc tấn công từ chối dịch vụ này về mặt lý thuyết sẽ không bao giờ dừng lại và các bên của kết nối sẽ ở trạng thái CLOSE-WAIT mãi mãi.

Bài báo của chúng tôi cải tiến phương pháp tấn công của Sơn để đề xuất kỹ thuật tấn công mới trên máy ảo và thử nghiệm tấn công trong mạng Docker. Các đề xuất tấn công thử nghiệm này được trình bày chi tiết trong các phần tiếp theo của bài báo.

E. Cấu trúc bài báo

Phần còn lại của bài báo được tổ chức như sau. Trong phần II, chúng tôi trình bày ngắn gọn tổng quan về các kiến thức liên quan. Chúng tôi tập trung vào giải thích về kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo và kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo. Để minh họa, chi tiết về các thí nghiệm tấn công từ chối dịch vụ dựa trên bão gói tin ACK được đề xuất trên các hệ thống máy chủ ảo, VMware và Docker, sẽ được mô tả trong phần III và phần IV. Dựa trên các thí nghiệm tấn công này, chúng tôi trình bày kết quả mô phỏng và các ý kiến thảo luận. Thêm nữa, trong phần V, chúng tôi đánh giá tính khả thi của kỹ thuật tấn công DoS ACK-Storm trên các hệ thống máy chủ ảo. Cuối cùng, Phần VI trình bày kết luận của bài báo này.

Bài báo này là phiên bản đầy đủ của bài báo trong Hội thảo quốc tế [4], tập trung vào nghiên cứu lỗ hổng bảo mật trong các hệ thống ảo và trình bày các giải pháp phòng chống ở mức đề xuất lý thuyết. Bài báo này chúng tôi phân tích cụ thể những thách thức của kỹ thuật ảo hoá máy chủ (mục I-A) để từ đó đánh giá được những đóng góp tích cực của bài báo đối với những hệ thống máy ảo được triển khai dịch vụ hiện nay (mục I-B). Bên cạnh đó, chúng tôi phân tích các kỹ thuật tấn công hiện nay đang được sử dụng để tấn công các hệ thống máy ảo dịch vụ trong mục I-C và mục I-D để đánh giá được các yếu tố của các hệ thống ảo đang bị tận dụng trong tấn công dịch vụ. Từ những phân tích bổ sung này, bài báo cũng mở rộng phần khuyến nghị để những nhà quản trị mạng đang triển khai các hệ thống dịch vụ trên các hệ thống ảo có những biện pháp phù hợp để hạn chế những tấn công đã được thử nghiệm trong bài báo.

II. CÁC NGHIÊN CỨU LIÊN QUAN

A. Kỹ thuật nền tảng

Trên thực tế, các hệ thống ảo hoá đang được sử dụng rất rộng rãi cho các ứng dụng cung cấp cho người dùng đầu cuối trên nền tảng Website. Các nhà phát triển ứng dụng đã và đang tiếp cận xây dựng ứng dụng phần mềm dựa trên nền tảng công nghệ Microservice nhằm tối ưu hoá nguồn lực phát triển và hiệu quả hoá tính mềm dẻo của các tính năng phần mềm[1]. Các chức năng của hệ thống đều được thiết kế chạy trên các dịch vụ (service) độc lập nhằm dễ xây dựng và thiết kế hệ thống phục vụ việc hiệu quả hoá trong mở rộng hệ thống trong tương lai. Một số ứng dụng điển hình đang sử dụng được phát triển trên nền tảng Microservice và hệ thống ảo hoá như *Viblo*[2] và *123Xe*[3]. Mỗi một tính năng lớn của hệ thống *Viblo* và *123Xe* được phát triển và cài đặt trên VMWare hoặc trên Docker và luôn được giám sát chặt chẽ quá trình vận hành của mỗi service.

Như vậy, để sử dụng các hệ thống máy ảo và Docker trong xây dựng hệ thống dịch vụ, các nhà phát triển phần mềm đã phải sử dụng những kỹ thuật, giao thức liên quan để kết nối các hệ thống máy ảo và Docker như: giao thức TCP được mô tả trong RFC-793 [11], Vmware Workstation, Hypervisor, Switch ảo - vSwitch, Bridge ảo - vBridge [23], [24], ... Những hệ thống máy ảo

[1]https://www.dotnetcoban.com/2020/04/reference-sample-projects.html

[2]https://viblo.asia/

[3]https://123xe.vn/

và Docker ban đầu được nghiên cứu để phát triển những dịch vụ đặt trong mạng nội bộ nhằm tối ưu hoá tài nguyên. Tuy nhiên, hiện nay thiết kế dịch vụ trên các hệ thống dịch vụ trên hệ thống ảo để cung cấp cho người dùng đầu cuối trở thành thông dụng và cần thiết. Do đó, việc nghiên cứu những khả năng xảy ra rủi ro đối với những hệ thống ảo thông dụng trở nên cấp bách nhằm đảm bảo an toàn cho hệ thống.

B. Một số nghiên cứu tấn công hệ thống ảo hoá

Công nghệ ảo hóa đóng một vai trò quan trọng trong việc xây dựng hệ thống điện toán đám mây dịch vụ (cloud computing) và hệ thống điện toán đám mây dành riêng (private cloud). Tuy nhiên, hiện vẫn có nhiều lỗ hổng trong việc triển khai ảo hóa và lớp ảo hóa (vBrigde, vSwitch) phải đối mặt với nhiều thách thức bảo mật khác nhau. Với sự tiện lợi của công nghệ ảo hóa mạng, một cơ sở hạ tầng mạng đơn lẻ có thể được chia thành nhiều kiến trúc mạng ảo. Điều này mang lại lợi ích cho một loạt các ứng dụng, bao gồm cả cơ sở hạ tầng điện toán đám mây. Trong bài báo [12], Bays $et\ al.$ đã thảo luận về một số thách thức và mối đe dọa chính liên quan đến an ninh mạng ảo, đồng thời trình bày các giải pháp tương ứng, cũng như các khía cạnh bảo mật chưa được tiếp cận khi thiết kế các dịch vụ liên quan đến hệ thống mạng ảo. Rõ ràng, trong môi trường hệ thống mạng ảo, bảo mật là rất quan trọng trong việc phát hiện các xâm nhập vào lớp mạng ảo. Do đó, Nathiya và cộng sự [13] đã đề xuất một hệ thống phát hiện xâm nhập kết hợp (hybrid intrusion detection system (H-IDS)) để giám sát an ninh trong một lớp mạng ảo, nhưng họ đã không triển khai và xác minh thử nghiệm.

Các vấn đề bảo mật trong mạng ảo hóa có thể được chia thành hai loại: Rủi ro bảo mật ảo hóa và tấn công bảo mật mạng ảo hóa. Tấn công máy ảo hóa phổ biến [14] bao gồm đánh cắp và giả mạo máy ảo, tấn công chuyển đổi máy ảo, tấn công rootkit dựa trên máy ảo (virtual-machine based rootkit - VMBR) và tấn công từ chối dịch vụ. Trong bài báo của Sơn và các đồng nghiệp [5] đã chứng minh rằng switch ảo (vSwitch) có thể truyền lại các gói TCP bị lỗi, điều đó chứng tỏ đặc điểm này sẽ bị lạm dụng cho các cuộc tấn công truyền lại gói tin bởi những kẻ tấn công nội bộ.

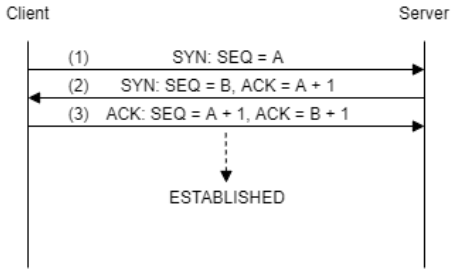
Mặt khác, thiết kế giao tiếp đơn giản giữa các máy ảo cũng tạo ra các lỗ hổng bảo mật như trong bài báo [15] đã nêu. Chính vì lẽ đó mà Elmrabet và cộng sự [16] đã đề xuất một kiến trúc bảo mật ba lớp mới, bao gồm switch ảo (vSwitch), tường lửa ảo (vFirewall) và VLAN, để ngăn chặn các cuộc tấn công vào các máy ảo thông qua các giao thức giao tiếp đơn giản giữa chúng. Sathya và cộng sự [17] đã đề xuất một mô hình đáng tin cậy cho bảo mật VM trong điện toán đám mây. Họ đã mã hóa VM image và sử dụng kỹ thuật snapshot và dịch vụ giám sát của bên thứ ba để cải thiện tính bảo mật, tính toàn vẹn và tính khả dụng của VM.

Dựa vào những phân tích, đánh giá về những nghiên cứu liên quan, chúng ta thấy được việc nghiên cứu để khai thác những lỗ hổng bảo mật trong hệ thống máy ảo hoá bao gồm cả Docker để đảm bảo bảo mật cho hệ thống dịch vụ triển khai trên máy ảo.

Giao thức điều khiển truyền vận – TCP TCP là giao thức truyền tin hướng kết nối, nó yêu cầu thiết lập kênh kết nối trước khi truyền dữ liệu. Thông qua TCP, các ứng dụng trên các máy chủ được kết nối mạng có thể giao tiếp với nhau, qua đó chúng có thể trao đổi dữ liệu hoặc gói tin. Giao thức này đảm bảo cung cấp dữ liệu đáng tin cậy cho người nhận. Hơn nữa, TCP có chức năng phân biệt giữa dữ liệu của nhiều ứng dụng (như dịch vụ Web, dịch vụ Email, v.v.) thực thi đồng thời trên cùng một máy chủ. Hoạt động của giao thức TCP được mô tả trong RFC-793 [11]. Ngày nay, TCP vẫn được sử dụng rộng rãi trong nhiều hệ thống máy chủ.

1) Bắt tay ba bước: Bắt tay ba bước được sử dụng trong giao thức TCP cho việc thiết lập kết nối. Giao thức TCP sử dụng mở thụ động (passive open), có nghĩa là một máy chủ được gán và lắng nghe trên một địa chỉ cổng trước khi một máy khách cố gắng kết nối đến nó thông qua địa chỉ đó. Một máy khách có thể bắt đầu một kết nối chủ động (active open) sau khi mở thụ động được thiết lập. Quá trình bắt tay ba bước diễn ra với ba bước như sau (xem Hình 1):

- Máy khách, máy muốn kết nối với máy chủ, sẽ gửi gói tin TCP đến máy chủ với giá trị ngẫu nhiên A cho số thứ tự (sequence number - SEQ) và bit cờ SYN được bật. Gói tin này được gọi là gói tin SYN (1).
- Sau khi máy chủ nhận được gói tin SYN,



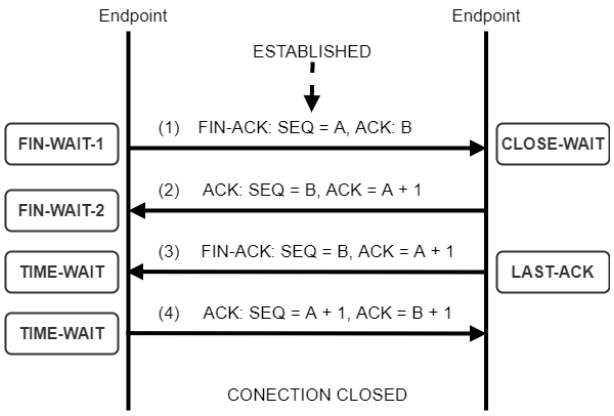
Hình 1: Quá trình Bắt tay ba bước trong TCP

nó phản hồi cho máy khách gói tin TCP có giá trị ngẫu nhiên B cho số thứ tự SEQ, số xác nhận (acknowledgment-ACK) được đặt giá trị bằng số thứ tự từ gói tin nhận được cộng thêm một (A + 1) và hai bit cờ SYN, ACK được bật. Gói tin này được gọi là gói tin SYN-ACK (2).

- Cuối cùng, máy khách gửi gói tin ACK (3), có số xác nhận được đặt giá trị bằng số thứ tự nhận được cộng thêm một (B + 1), giá trị số thứ tự được đặt bằng giá trị xác nhận nhận được (A + 1) và chỉ có bit cờ ACK được bật. Đây là gói tin ACK có nhiệm vụ hoàn thành quá trình thiết lập cho kết nối TCP.

2) Bắt tay bốn bước: Bắt tay bốn bước được sử dụng để kết thúc một kết nối TCP với việc kết thúc độc lập của mỗi bên trong kết nối. Quá trình này diễn ra như sau:

- Khi một bên X trong một kết nối TCP muốn kết thúc kết nối. nó gửi tới bên còn lại Y của kết nối một gói tin FIN, với bit cờ FIN được bật, số thứ tự phân đoạn SEQ (A), và số xác nhận ACK (B) dựa trên trạng thái hiện tại của kết nối TCP đó. Sau đó, bên X chuyển sang trạng thái FIN-WAIT-1.
- Khi thiết bị đầu cuối Y nhận gói tin FIN, nó giải phóng bộ nhớ đệm, và phản hồi một gói tin ACK, với số xác nhận ACK được đặt giá trị bằng số thứ tự phân đoạn nhận được cộng thêm một (A+1) và số thứ tự phân đoạn được đặt bằng số xác nhận nhận được (B). Nó cũng đồng thời chuyển sang trạng thái CLOSE-WAIT. Sau khi nhận gói tin ACK từ bên Y, bên X chuyển sang trạng thái FIN-WAIT-2 từ trạng thái FIN-WAIT-1. Kết nối TCP từ X sang Y đã kết thúc, tuy nhiên kết nối TCP từ Y sang X vẫn đang mở, đây là kết nối "nửa mở" hay kết nối "nửa đóng".
- Để kết nối TCP kết thúc hoàn toàn, thiết bị đầu cuối Y gửi gói tin FIN-ACK để kết thúc kết nối TCP từ Y sang X và đợi một gói tin xác nhận (gói tin ACK) từ X.
- Cuối cùng, khi thiết bị đầu cuối X nhận được gói tin FIN từ Y, X gửi gói tin ACK cuối cùng của kết nối TCP này và X chuyển sang trạng thái CLOSED. Lúc này, kết nối TCP đã kết thúc hoàn toàn (ở cả 2 bên X và Y).



Hình 2: Quá trình Bắt tay bốn bước trong TCP

C. Vmware Workstation

1) Máy ảo - VMs: Máy ảo là các phần mềm máy tính cung cấp các chức năng giống như các máy tính vật lý. Các máy ảo dựa trên kiến trúc máy tính vật lý nhưng chúng hoạt động như các hệ thống máy tính vật lý độc lập. Với máy ảo, người dùng có thể thực thi các phần mềm yêu cầu các môi trường khác nhau mà không cần lo lắng tới việc xảy ra các xung đột trong cùng một thời điểm (ví dụ như: các xung đột liên quan tới phiên bản hoặc xung đột về môi trường hệ điều hành .v.v.). Các máy ảo đem tới nhiều lợi ích cho người sử dụng và các doanh nghiệp như là: Giảm thiểu chi phí phần cứng; Tăng tốc quá trình triển khai hệ thống máy tính và máy chủ; Tiết kiệm năng lượng; Gia tăng hiệu quả hoạt động công nghệ thông tin và các lợi ích khác.

2) Hypervisor: Hypervisor, hoặc còn gọi là trình quản lý máy ảo (VMM), có thể là phần cứng, phần mềm hoặc chương trình cơ sở (firmware) cung cấp khả năng ảo hóa. Một máy chủ ảo cho phép một máy chủ, máy có máy chủ ảo hoạt động trên nó, hỗ trợ nhiều máy ảo khách bằng cách chia sẻ các tài nguyên của nó như bộ nhớ và vi xử lý, v.v. Các tài nguyên đã được phân bổ cho mỗi máy ảo, máy chủ ảo cung cấp và quản lý việc lập lịch

sử dụng các tài nguyên máy ảo dựa trên các tài nguyên vật lý. Máy chủ ảo có hai loại: loại 1, "bare metal", thực thi trực tiếp trên phần cứng máy chủ lưu trữ, giống như một hệ điều hành, trong khi loại 2, "hosted", hoạt động như một phần mềm trên hệ điều hành, tương tự như một ứng dụng. Việc lựa chọn loại máy chủ ảo dựa trên mục đích, nhu cầu của người dùng và doanh nghiệp.

3) Switch ảo - vSwitch: Switch ảo là một ứng dụng phần mềm cho phép giao tiếp giữa các máy ảo, giữa máy vật lý và máy ảo. Nó điều khiển giao tiếp trên mạng một cách thông minh bằng cách đảm bảo tính toàn vẹn của hồ sơ máy ảo, bao gồm cài đặt mạng và bảo mật, kiểm tra các gói tin dữ liệu trước khi điều hướng chúng đến đích. Một switch ảo là hoàn toàn ảo và có thể kết nối với card giao diện mạng (NIC). Các vSwitch hợp nhất các switch vật lý thành một switch logic duy nhất. Điều này giúp tăng băng thông và tạo mạng lưới hoạt động giữa máy chủ và các switch. vSwitch cũng giúp dễ dàng triển khai và di chuyển các máy chủ ảo, cho phép nhà quản trị mạng quản lý switch ảo được triển khai thông qua một máy chủ ảo và dễ dàng triển khai chức năng mới, có thể liên quan đến phần cứng hoặc chương trình cơ sở (firmware).

4) Mạng ảo - Mạng ảo hóa: Mạng ảo hóa là một phương pháp phân chia băng thông có sẵn thành các kênh để kết hợp các tài nguyên có sẵn trong mạng. Mỗi kênh có thể được chỉ định (hoặc được chỉ định lại) cho một máy chủ hoặc thiết bị cụ thể trong thời gian thực và được bảo mật độc lập. Ý tưởng chính của mạng ảo hóa là ảo hóa che giấu sự phức tạp thực sự của mạng bằng cách chia nó thành các phần có thể quản lý, giống như ổ cứng được phân vùng giúp quản lý tệp dễ dàng hơn. Mỗi máy đăng ký chia sẻ quyền truy cập vào tất cả các tài nguyên trên mạng từ một máy tính đơn.

D. Docker

1) Docker Engine - Docker Daemon: Docker Engine, hay còn gọi là Docker Daemon, là một dịch vụ chạy trong nền, quản lý mọi thứ cần thiết để thực thi và tương tác với các Docker Container trên hệ điều hành máy chủ. Nó được sử dụng để thực thi các Docker Container chứa tất cả các phụ thuộc cần cho ứng dụng bên trong. Docker Engine cho phép các ứng dụng được đóng gói, có thể chạy một cách nhất quán trên mọi cơ sở hạ tầng. Docker Daemon giao tiếp trực tiếp với hệ điều hành máy chủ và biết cách phân phối tài nguyên cho các Docker Container đang chạy. Docker Engine đảm bảo mỗi Container được cách ly với cả hệ điều hành máy chủ cũng như các Container khác. Hiểu một cách đơn giản, nó thay thế cho Hypervisor trong VMWare.

2) Docker Container Image: Docker Container Image, hay Docker Image, là một gói phần mềm nhẹ, độc lập, có thể thực thi được. Nó bao gồm: mã lệnh, thời gian chạy (run time), công cụ hệ thống, thư viện hệ thống và các tệp cài đặt - tất cả những thứ cần thiết để thực thi ứng dụng. Có nhiều Docker Image có sẵn có thể được sử dụng để xây dựng lại Docker Image mới hoặc để triển khai các Docker Container.

3) Docker Container: Docker Container là một thể hiện khi thực thi của Docker Image. Tuy nhiên, chúng ta có thể tùy chỉnh các Docker Container. Nhiều Container có thể chạy trên cùng một máy tại cùng một thời điểm và chia sẻ nhân (kernel) của hệ điều hành. Các Container hoạt động như các tiến trình độc lập trong không gian người dùng (userspace). Các container chiếm ít không gian hơn máy ảo (VMs), có thể xử lý nhiều ứng dụng hơn và yêu cầu ít hơn các máy ảo và các hệ điều hành thực.

4) Docker Network: Nguyên lý hoạt động của mạng Docker là hướng ứng dụng. Sự cách ly trong mạng Docker đạt được bằng cách sử dụng không gian tên mạng. Thông thường, các dịch vụ nhận IP riêng và ánh xạ tới nhiều container. Các dịch vụ được thực hiện khi Container chú trọng hơn vào khám phá dịch vụ tích hợp. Vì quy mô Container trên một máy chủ có thể chạy tới hàng trăm Container, nên mạng máy chủ phải có khả năng mở rộng rất lớn.

5) Bridge ảo - vBridge: Một bridge ảo (vBridge), có tên gọi khác là Network bridge hoặc Linux bridge,là một phần mềm được sử dụng để hợp nhất hai hoặc nhiều phân đoạn mạng. Nó hoạt động như một switch ảo (vSwitch) và hoạt động một cách trong suốt (không ảnh hưởng đến quá trình truyền tin) [23] [24]. Trong Docker container được xây dựng với Image cơ sở là Linux, mạng Docker được quản lý bởi vBridge thay vì vSwitch như trên VMware Workstation.

E. Chuyển đổi địa chỉ mạng và chuyển đổi địa chỉ cổng (NAT, PAT)

1) Chuyển đổi địa chỉ mạng - Network Address Translation (NAT): Chuyển đổi địa chỉ mạng (NAT) là một kỹ thuật cho phép một hoặc nhiều địa chỉ IP nội bộ (mạng cục bộ) được chuyển đổi thành một hoặc nhiều địa chỉ IP công cộng (bên ngoài). Kỹ thuật này giúp cho một thiết bị trong mạng cục bộ/mạng riêng có thể kết nối với mạng công cộng (Internet). NAT chịu trách nhiệm truyền các gói tin từ lớp mạng này sang lớp khác trong cùng một mạng. NAT sẽ thay đổi địa chỉ IP bên trong gói tin. Sau đó, các gói tin sẽ di chuyển qua các bộ định tuyến và thiết bị mạng. Ngược lại, khi gói tin được truyền từ Internet (mạng công cộng) trở lại NAT, NAT thực hiện nhiệm vụ thay đổi địa chỉ đích thành địa chỉ IP tương ứng bên trong mạng cục bộ và gửi nó đến thiết bị đầu cuối tương ứng. Hơn nữa, NAT có thể hoạt động như một tường lửa. Nó giúp người dùng bảo mật thông tin IP máy tính trong mạng cục bộ. Cụ thể, nếu máy tính gặp sự cố khi kết nối với Internet, địa chỉ IP trong mạng công cộng (được thiết lập cấu hình trước đó) được hiển thị thay vì địa chỉ IP trong mạng cục bộ.

2) Chuyển đổi địa chỉ cổng - Port Address Translation (PAT): Chuyển đổi địa chỉ cổng (PAT) là kỹ thuật mở rộng của NAT, có thể giúp nhiều thiết bị trên mạng cục bộ/mạng riêng kết nối với mạng công cộng bằng cách ánh xạ địa chỉ IP cục bộ của chúng sang một địa chỉ IP công cộng duy nhất với các địa chỉ cổng cụ thể. Với mỗi cặp "địa chỉ IP cục bộ:cổng cục bộ" được ánh xạ tới địa chỉ IP công cộng với một địa chỉ cổng cụ thể, nhiều thiết bị có thể giao tiếp với Internet qua các cổng tương ứng được cung cấp cho chúng. Kỹ thuật này có thể bảo tồn địa chỉ IP nhưng số lượng địa chỉ cổng không vô hạn, chỉ có 65.536 cổng nên có thể có tối đa (trên lý thuyết) là 65.536 địa chỉ tại một thời điểm cho mỗi địa chỉ toàn cầu (địa chỉ IP công cộng). Nếu kẻ tấn công có thể chiếm tất cả 65.536 cổng, thì máy chủ dịch vụ sẽ rơi vào trạng thái “cạn kiệt cổng" và không thể thực hiện giao tiếp giữa các thiết bị cục bộ và mạng công cộng.

III. ĐỀ XUẤT TẤN CÔNG TỪ CHỐI DỊCH VỤ DỰA TRÊN BÃO GÓI TIN ACK TRÊN VMWARE WORKSTATION

A. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo

1) Môi trường thực nghiệm (xem Hình 3):

a) Máy chủ ảo: Chúng tôi sử dụng VMware Workstation 14.1.1 có cài đặt Windows 10 64bit làm máy chủ ảo B*.

b) Các máy vật lý: Máy khách A và máy chủ ảo B có cùng cấu hình như sau: Window 10 64bit, Chip Inter® Core™ i7-6700 CPU @ 3.40GHz, RAM 8GB, card mạng 100Mbps, bộ chuyển đổi Ethernet gắn với PCI-E bus. Máy tấn công C sử dụng Windows 10 có cài đặt Python 3.7.3.

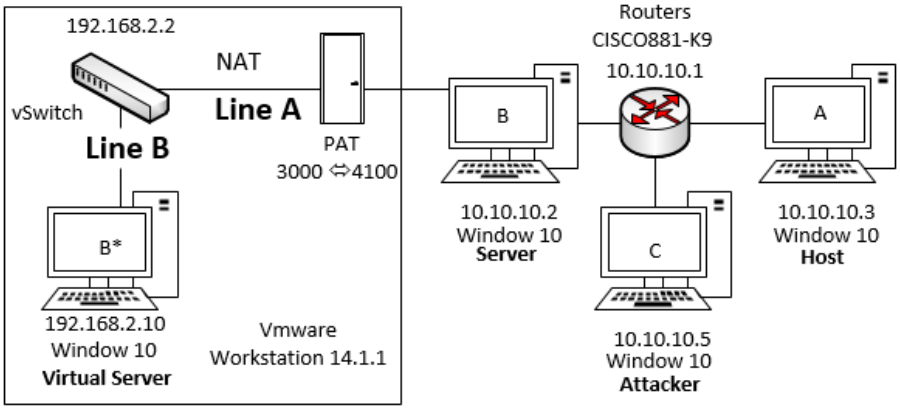
c) Router: Chúng tôi sử dụng router thay cho hub trong thí nghiệm của tác giả Sơn vì trong thực thế các doanh nghiệp sử dụng router để thiết lập mạng LAN hoặc để kết nối Internet. Loại router sử dụng trong các thí nghiệm của chúng tôi là Cisco 800 Series Routers CISCO881-K9.

d) Thu thập gói tin: Trong các thực nghiệm, các gói tin trên đường truyền giữa máy chủ vật lý B và vSwitch - Line A, và trên đường truyền giữa máy chủ ảo B* và vSwitch - Line B được bắt để phân tích. Với việc tiến hành bắt các gói tin trên cả Line A và Line B, chúng tôi kiểm chứng tính khả thi của kỹ thuật tấn công đồng thời đề xuất các biện pháp phòng chống phù hợp.

2) Thực hiện tấn công: Kết nối TCP bình thường được thiết lập bằng đoạn mã python (v3.7.3) sử dụng lập trình socket. Trong thí nghiệm này, chúng tôi chỉ cho máy khách A mở kết nối với máy chủ ảo B* bằng quá trình bắt tay 3 bước và lấy gói tin ACK cuối cùng từ kết nối TCP đã thiết lập thành công. Máy tấn công C sử dụng các số xác nhận (ACK number) và số thứ tự phân đoạn (SEQ number) của gói tin ACK cuối cùng này để tạo gói tin giả có địa chỉ nguồn là 1 máy trong máy khách A và máy B*, địa chỉ IP đích là máy còn lại bằng scapy[4]. Sau đó, chúng tôi gửi các gói tin đến máy có địa chỉ IP đích tương ứng.

a) Trên Line A: Chúng tôi bắt được khoảng 55,000 gói tin ACK truyền lại (retransmitted ACK) trong 60 giây khi cuộc tấn công từ chối dịch vụ dựa trên bão gói tin ACK đang diễn ra. Kết quả này tương đồng với kết quả trong thí

[4]https://scapy.net/



Hình 3: Mô hình thí nghiệm trên VMware Workstation - Ba máy vật lý như trên được kết nối với nhau qua Router cisco loại Routers CISCO881-K9, máy C là máy tấn công và có thể nghe lén, và tiêm các gói tin vào kết nối TCP giữa máy khách A và máy chủ ảo hóa B.

nghiệm trước đó của tác giả Abramov [3] và của tác giả Sơn [5], nhưng số lượng gói tin ít hơn là do băng thông của Bộ chuyển đổi Ethernet trong thí nghiệm của chúng tôi nhỏ hơn. Tuy nhiên, trình quản lý Window Taskmanager vẫn hiển thị băng thông sử dụng bởi VMware NAT services là 0.6Mbps. Kết quả này lớn hơn nhiều so với 120 byte (hai gói tin giả mạo) được gửi bởi máy tấn công C.

b) Trên Line B: Chỉ gói tin ACK giả đầu tiên mà chúng tôi tạo ra được chuyển tiếp đến máy ảo B* thông qua vSwitch. Không có gói tin ACK truyền lại nào được chuyển đến máy ảo B*.

B. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo

1) Môi trường thực nghiệm: Chúng tôi sử dụng cùng một môi trường với thí nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo được mô tả ở trên (trong phần III-A).

2) Thực hiện tấn công: Thực hiện tấn công tương tự với thực nghiệm tấn công từ chối dịch vụ dựa trên bão gói tin ACK bằng gói tin ACK giả mạo như mô tả ở trên, nhưng các gói tin FIN-ACK giả mạo được sử dụng thay vì các gói tin ACK giả mạo. Điều đó có nghĩa chỉ cần bật thêm bit cờ FIN và giữ nguyên tất cả các điều kiện khi tiến hành thực nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo trong phần III-A.

3) Phân tích: Khi mỗi bên của kết nối TCP nhận được các gói tin FIN-ACK giả mạo từ máy tấn công C, các bên sẽ phản hồi với các gói tin ACK truyền lại không hợp lệ và cuộc tấn công từ chối dịch vụ dựa trên bão gói tin ACK được kích hoạt thành công, gây ngưng trệ dịch vụ của toàn bộ hệ thống mạng.

a) Trên Line A: Chúng tôi đã bắt được khoảng 86,000 gói tin ACK truyền lại trong 60 giây (khoảng 1434 gói tin mỗi giây) trên Line A trong khi cuộc tấn công từ chối dịch vụ đang diễn ra. Kết quả này tương tự với kết quả trong các thử nghiệm trước đây của tác giả Sơn [5] nhưng số lượng gói tin ít hơn do băng thông bộ chuyển đổi Ethernet nhỏ hơn. Tuy nhiên, trình quản lý tác vụ vẫn hiển thị băng thông được sử dụng bởi các dịch vụ VMware NAT là 0,6Mbps. Kết quả này lớn hơn nhiều so với 120 byte (hai gói tin) được gửi bởi kẻ tấn công C. Chúng tôi tiếp tục thử nghiệm trong 24 giờ và nó vẫn hoạt động. Điều này chứng tỏ rằng giả thuyết của tác giả Sơn [5] là chính xác. Kết nối TCP bị kẹt trong trạng thái CLOSE-WAIT trong khi tiến trình thiết lập kết nối TCP còn hoạt động. Vì cơn bão ACK được kích hoạt bởi một vài gói tin FIN-ACK giả mạo có thể tồn tại mãi mãi, kẻ tấn công hoàn toàn có thể nâng cao tấn công DoS lên thành tấn công DDoS (distributed denial-of-service attack) lên máy chủ vật lý B và máy ảo B*. Nếu kẻ tấn công có thể thực hiện một cuộc tấn công DDoS với tất cả các cổng có sẵn, khoảng 65,535 cổng, kẻ tấn công có thể tạo một cuộc tấn công DDoS với 94 triệu gói tin mỗi giây (với lưu lượng mạng 40 Gbps) và đồng thời gây ra hiện tượng "cạn kiệt cổng".

b) Trên Line B: Tương tự như kết quả ở thực nghiệm trên, trong thực nghiệm tấn công DoS dựa

trên bão gói tin ACK, chỉ có gói tin FIN-ACK giả mạo đầu tiên mà chúng tôi tạo ra từ máy tấn công C được chuyển tiếp đến máy ảo B* thông qua vSwicth. Tuy nhiên, không có gói tin ACK truyền lại nào được chuyển đến máy ảo B*, kết quả tương tự với thí nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo được đề xuất lần đầu. Chúng tôi giả định rằng vSwitch phản hồi tất cả các gói tin ACK được truyền lại thay vì máy ảo B* miễn là không có gói tin RST nào được gửi giữa máy khách A và máy ảo B*. Để chứng minh điều này, chúng tôi thử tạm ngưng hoạt động ("suspend") máy ảo B*, cuộc tấn công vẫn diễn ra. Chúng tôi tiếp tục thử nghiệm thêm 4 giờ nữa và cơn bão ACK này không có dấu hiệu dừng lại. Khi chúng tôi khôi phục hoạt động ("resume") cho máy ảo B*, không có gì bất thường xảy ra. Đây có thể là một tính năng của VMware Workstation, vSwitch nhằm ngăn chặn tất cả các gói tin không hợp lệ vào máy ảo bên trong nó. Tuy nhiên, switch ảo lại không có khả năng ngăn chặn, kết thúc cuộc tấn công DoS ACK-Storm này, thiếu các biện pháp bảo mật [29].

IV. ĐỀ XUẤT THỬ NGHIỆM TẤN CÔNG TỪ CHỐI DỊCH VỤ DỰA TRÊN BÃO GÓI TIN ACK TRÊN DOCKER

A. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo

1) Môi trường thực nghiệm:

a) Trình quản lý ảo hóa: Chúng tôi sử dụng Docker Desktop v2.2.0.5 với Docker Engine v19.03.8, Docker Compose v1.25.4. Chúng tôi xây dựng một Docker Image mới dựa vào Image cơ sở là Ubuntu 18.04 và cài đặt Htop[5] để theo dõi hoạt động của Container. Chúng tôi cũng sử dụng TCPdump để theo dõi luồng dữ liệu mạng và sử dụng Nginx cho máy chủ trên Container B*.

b) Các máy vật lý: Máy khách A and máy chủ vật lý B có cùng cấu hình như sau: Window 10 64bit, Chip Inter® Core™ i7-6700 CPU @ 3.40GHz, RAM 8GB, Bộ chuyển đổi giao diện mạng Ethernet 100Mbps gắn với PCI-E bus. Máy tấn công C sử dụng Windows 10 có cài đặt Python 3.7.3.

c) Router: Router được sử dụng để thiết lập mạng LAN. Router sử dụng trong thực nghiệm này là Cisco 800 Series Routers CISCO881-K9.

[5]https://hisham.hm/htop/

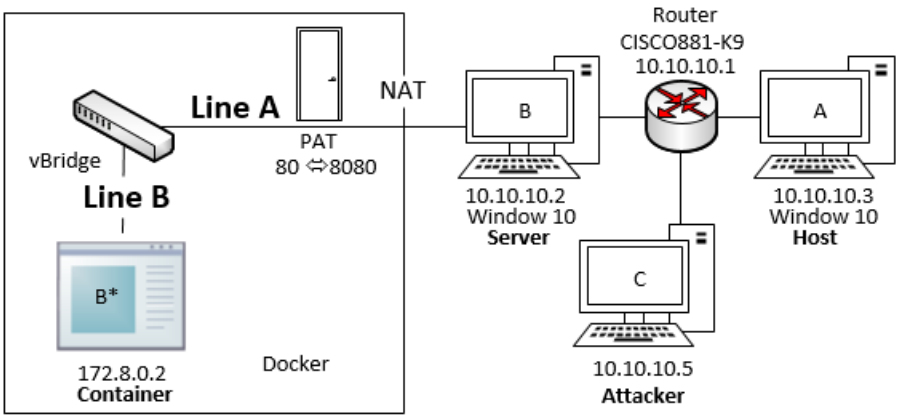
d) Thu thập gói tin: Thực hiện tương tự như các thí nghiệm trên môi trường VMware Workstation, chúng tôi bắt các gói tin trên đường truyền giữa chủ thật B và Bridge - Line A, và trên đường truyền giữa máy chủ ảo B* và vBridge - Line B.

2) Thực hiện tấn công: Với kết nối TCP được tạo bởi trình duyệt Chrome mở kết nối với Nginx trên Container B* và thực hiện tương tự như trong thí nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo trên môi trường ảo VMware Workstation. Chúng tôi lấy gói tin ACK cuối cùng trong kết nối TCP giữa máy khách A và Docker Container B* để có được số xác nhận (ACK) và số thứ tự phân đoạn (SEQ) hợp lệ. Tiếp đến, chúng tôi sử dụng scapy để tạo một cặp gói tin ACK giả mạo với địa chỉ nguồn và đích tương ứng với máy khách A và máy chủ vật lý B. Sau đó, chúng tôi sử dụng máy tấn công C, gửi các gói tin này tới các máy đích tương ứng.

3) Phân tích: Khi các bên trong kết nối TCP này nhận được gói tin ACK giả mạo, chúng gửi lại các gói tin ACK truyền lại không hợp lệ, cuộc tấn công từ chối dịch vụ được kích hoạt. Cuộc tấn công này kết thúc bởi máy khách A (xem Hình 4) sau 1 giây và "timeout", quá thời gian chờ, sau đó 30 giây.

a) Trên Line A: Chúng tôi bắt được khoảng 550 gói tin ACK truyền lại trong một giây trong khi cơn bão ACK đang diễn ra. 10 giây sau đó, Container B* gửi gói tin RST-ACK cho máy khách A. Kết nối TCP "timeout" sau 30 giây. Cuộc tấn công không duy trì lâu nhưng vẫn có một lượng lớn gói tin ACK truyền lại được sinh ra bởi kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo. Điều này là do tính năng của Docker hoặc của Nginx được trang bị để chống các cuốc tấn công từ chối dịch vụ khi Container nhận được nhiều gói tin ACK truyền lại với cùng số ACK/SEQ. Kết quả này không như mong đợi nhưng vẫn chứng minh lỗ hổng trong TCP, điều đã được tác giả Abramov chỉ ra trong [3], khi kết nối TCP nhận được gói tin có số xác nhận ACK "chưa được gửi" (not-yet-sent).

b) Trên Line B: Kết quả thu được tương tự như trong thực nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo trên VMware Workstation, chỉ duy nhất gói tin ACK giả mạo đầu tiên được điều hướng tới Docker Container B* qua vBridge. Không có gói tin



Hình 4: Mô hình thử nghiệm trên Docker- Ba máy tính vật lý như trên được kết nối bằng Cisco 800 Series Routers CISCO881-K9, máy tính C là máy tấn công có khả năng nghe lén và "tiêm" các gói tin vào kết nối TCP giữa khách A và máy chủ B. Docker Container B* sử dụng Nginx cho máy chủ với cổng cục bộ là 80, cổng công cộng là 8080.

ACK truyền lại nào được điều hướng tới Docker Container B*. Điều này cho thấy vBridge phản hồi lại toàn bộ gói tin ACK truyền lại không hợp lệ thay cho Docker Container B*, hoạt động này giống như hoạt động của vSwitch trên VMware Workstation.

B. Kỹ thuật tấn công từ chối dịch vụ dựa trên bão gói tin ACK bằng gói tin FIN-ACK giả mạo

1) Môi trường thực nghiệm: Chúng tôi sử dụng môi trường thực nghiệm tương tự như thực nghiệm kỹ thuật tấn công DoS ACK-Storm trên Docker đã được mô tả ở phía trên.

2) Thực hiện tấn công: Chúng tôi tiến hành tương tự như thực nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo trên Docker nhưng sử dụng một cặp gói tin FIN-ACK giả mạo thay cho cặp gói tin ACK giả mạo. Các điều kiện khác của thực nghiệm được giữ nguyên.

3) Phân tích: Khi mỗi bên nhận được các gói tin FIN-ACK giả mạo, chúng phản hồi với các gói tin ACK truyền lại không hợp lệ và cơn bão ACK bắt đầu.

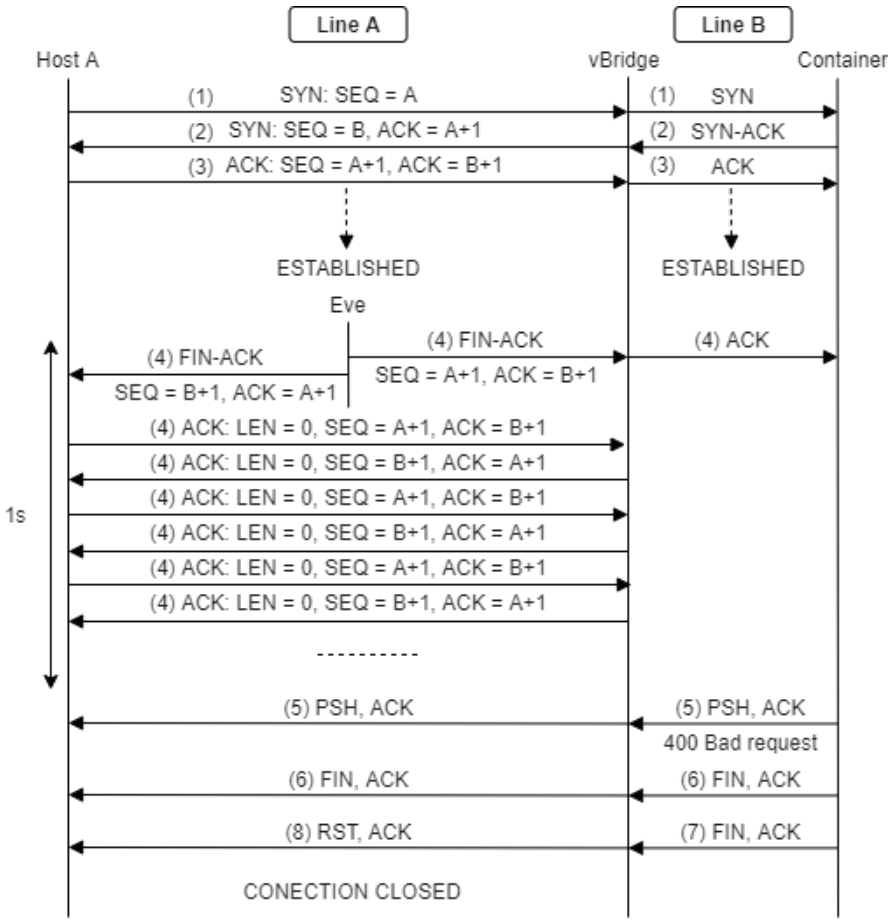
a) Trên Line A: Chúng tôi bắt được một vài gói tin ACK truyền lại giữa máy khách A và Container B* trước khi phía Container B* ngừng phản hồi.

b) Trên Line B: Kết quả thu được tương tự với thực nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo, chỉ có gói tin ACK giả mạo đầu tiên được điều hướng tới Container B* qua vBridge. Không có gói tin ACK truyền lại nào được điều hướng tới Container B*.

4) Đề xuất giải pháp tấn công từ máy trung gian C: Chúng tôi đã cố gắng thay thế các gói tin ACK đơn giản để tấn công bằng các gói tin HTTP. Tuy nhiên, chúng tôi đã nhận được kết quả tương tự như mục IV-A và mục IV-B. Do kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo [5] dường như không hoạt động với kết nối TCP được tạo bởi các trình duyệt như Chrome hoặc Firefox với Nginx, chúng tôi đề xuất một phương pháp tấn công mới dựa trên nó: máy tấn công C tự tạo kết nối TCP giả với máy chủ B, gửi gói tin FIN-ACK giả đến máy chủ B sau khi bắt tay ba bước hoàn thành và liên tục gửi các gói tin ACK truyền lại giả có cùng định dạng như các gói tin ACK truyền lại của kỹ thuật tấn công DoS ACK-Storm bằng gói tin ACK giả mạo. Máy chủ B, cụ thể là vBridge, sẽ phản hồi lại các gói tin này và bị kẹt trong trạng thái CLOSE-WAIT. Với kết nối TCP đơn giản, trạng thái CLOSE-WAIT không có "timeout" [11]. Vì vậy, chúng tôi cho rằng máy chủ B có thể bị mắc kẹt trong trạng thái CLOSE-WAIT mãi mãi (có nghĩa là tấn công từ chối dịch vụ trên môi trường ảo Docker đã thành công). Chi tiết về phương pháp tấn công này sẽ được mô tả trong mục IV-C.

C. Kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo và kết nối giả

1) Môi trường thực nghiệm: Chúng tôi sử dụng cùng một môi trường với thí nghiệm kỹ thuật tấn



Hình 5: Kết nối TCP trong quá trình thí nghiệm trên Docker với gói tin ACK giả mạo

công DoS ACK-Storm trên Docker được mô tả ở trên (xem Hình 5) nhưng không có máy khách A.

2) Thực hiện tấn công: Chúng tôi sử dụng scapy (2.4.3) để tạo kết nối TCP giả với Container B* bằng quá trình bắt tay ba bước như sau:

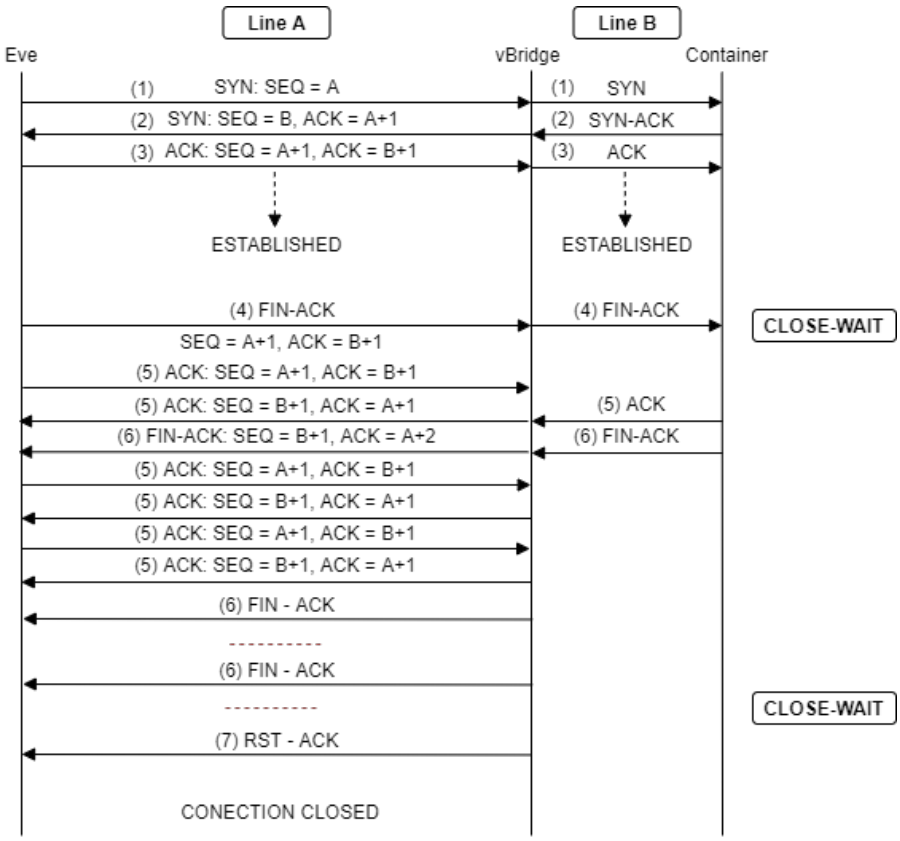
- Sử dụng scapy tạo một gói tin SYN giả bằng cách thủ công với địa chỉ IP đích là Container B*, địa chỉ IP nguồn là máy tấn công C, số xác nhận ACK là 0, số thứ tự phân đoạn SEQ là tùy ý và gửi đến Container B*.
- Bắt gói tin phản hồi SYN-ACK từ Container B* sau đó tạo gói tin ACK giả dựa vào gói tin SYN-ACK (sử dụng số thứ tự ACK và số thứ tự phân đoạn SEQ).
- Gửi gói tin ACK giả tới Container B* và kết nối TCP được thiết lập.

Tiếp đến, chúng tôi tạo một gói tin FIN-ACK giả giống như trong thí nghiệm trước đó và tạo một gói tin ACK truyền lại giống như việc máy tấn công C nhận được gói tin FIN-ACK giả tương tự như Container B*. Sau đó, chúng tôi tiếp tục thực hiện các bước sau:

- Gửi gói tin FIN-ACK giả tới Container B*.
- Gửi gói tin ACK truyền lại giả tới Container B*.
- Trì hoãn 1 giây và thực hiện lại bước 2.

3) Phân tích: Khi Container B* nhận được gói tin FIN-ACK giả, nó sẽ chuyển sang trạng thái CLOSE-WAIT và phản hồi với các gói tin ACK truyền lại không hợp lệ.

a) Trên Line A: Bất cứ khi nào vBridge nhận được gói tin ACK truyền lại giả từ máy tấn công C, nó sẽ phản hồi với các gói tin ACK truyền lại không hợp lệ. Trong khi máy tấn công C duy trì việc gửi gói tin ACK truyền lại giả đến Container B*, thì Container B* bị mắc kẹt ở trạng thái CLOSE-WAIT. Nhưng cuộc tấn công không kéo dài quá lâu như trong thí nghiệm kỹ thuật tấn công DoS ACK-Storm bằng gói tin FIN-ACK giả mạo [5]. Sau khoảng 10 phút, Container B* gửi gói tin RST để đóng kết nối TCP. Trong thử nghiệm, chúng tôi có bắt được một số gói tin FIN-PSH-



Hình 6: Kết nối TCP trong quá trình thí nghiệm sử dụng gói tin FIN-ACK giả trên Docker với các gói tin ACK truyền lại giả từ máy tấn công C

ACK nhưng không có dấu hiệu của việc ngắt kết nối cho đến khi Container B* đột ngột gửi gói tin RST. Đây có thể là tính năng của Container B* khi không nhận được gói tin phản hồi và vBridge khi nhận được quá nhiều gói tin ACK được truyền lại với cùng số ACK/SEQ.

b) Trên Line B: Chỉ các gói tin hợp lệ được chuyển đến Container B*, giống như kết quả của thí nghiệm kỹ thuật tấn công DoS ACK-Storm. Không có gói tin ACK truyền lại (giả) nào được chuyển đến Container B*. Điều đó có nghĩa là vBridge phản hồi tất cả các gói tin ACK được truyền lại thay vì Container B* miễn là không có yêu cầu RST nào được gửi giữa máy khách A và Container B*.

V. Thảo luận

A. Tính khả thi

Từ kết quả của các thí nghiệm trên, chúng tôi kết luận rằng kỹ thuật tấn công DoS ACK-Storm là khả thi với các kết nối TCP cơ bản trong môi trường ảo hóa. Dựa trên lỗ hổng được phát hiện bởi tác giả Abramov [3] được mô tả trong RFC 793 [11] và dựa trên ý tưởng phát triển phương pháp tấn công bằng gói tin FIN-ACK của tác giả Sơn [5], máy tấn công có thể thực hiện cuộc tấn công bằng cách tạo kết nối giả và gửi các gói tin giống như kết nối TCP bình thường đang bị tấn công từ chối dịch vụ dựa trên bão gói tin ACK bằng gói tin FIN-ACK giả mạo.

Trong các thí nghiệm của chúng tôi, kỹ thuật tấn công này là khả thi ngay cả với các môi trường ảo hóa như Docker sử dụng Nginx cho máy chủ. Tuy nhiên, trong cấu hình mặc định của Nginx, mỗi trạng thái của kết nối có cài đặt thời gian "timeout" nên phương pháp tấn công sử dụng gói tin ACK bị chặn và nguy cơ bị kẻ tấn công tấn công bằng gói tin FIN-ACK cũng bị hạn chế. Tuy nhiên, kẻ tấn công C có thể sử dụng một đoạn mã python đơn giản với thư viện scapy để tạo nhiều kết nối đến máy chủ trên Docker Container sử dụng Nginx để chiếm cổng trong vòng 10 phút và có thể gây ra sự cố, được gọi là “cạn kiệt cổng", với vBridge khi kết hợp với mạng botnet.

Thời gian, tuy chỉ khoảng 10 phút, không quá dài, nhưng thiệt hại cho các doanh nghiệp sẽ không nhỏ mặc dù chi phí thực hiện cuộc tấn công không quá cao. Với kỹ thuật tấn công này, kẻ tấn công không cần khả năng nghe lén kết nối TCP nhưng vẫn có thể chiếm cổng.

B. Giải pháp phòng chống

Như chúng tôi đã đề cập ở trên, trong phần III và phần IV, vSwitch (hoặc vBridge) phản hồi (hoặc bỏ qua) các gói tin không hợp lệ thay máy chủ ảo (hoặc Container) và chỉ điều hướng gói tin hợp lệ đến máy chủ ảo. Để đạt được điều này, hệ thống cần có một module giám sát các gói tin trước vSwitch (hoặc vBridge) để xác định gói tin không hợp lệ hay không. Gói tin không hợp lệ, gói tin giả mạo là những gói tin trùng ACK number, trùng SEQ number, trùng trạng thái FIN-ACK để bật chế độ CLOSE-WAIT của máy chủ ảo. Bộ giám sát này chỉ truyền gói tin hợp lệ đến máy chủ ảo để máy chủ ảo xử lý và phản hồi đến máy khách hợp lệ. Do đó, máy chủ ảo sẽ không nhận được gói tin nào từ máy khách trong suốt cuộc tấn công từ chối dịch vụ dựa trên bão gói tin ACK.

Để phòng chống kỹ thuật tấn công này có thể sử dụng một bộ đếm thời gian chạy trên một luồng khác bên cạnh vSwitch (hoặc vBridge) để đếm thời gian không phản hồi từ máy khách và đóng kết nối đó khi “timeout". Có nghĩa là bộ đếm thời gian sẽ xác định thời gian cần thiết để máy chủ ảo thoát khỏi trạng thái CLOSE-WAIT mà không duy trì mãi mãi trạng thái này, từ đó hệ thống dịch vụ sẽ không bị ngưng trệ khi nhận được gói tin giả mạo ngắt kết nối từ kẻ tấn công nội bộ.

Ngoài ra, giải pháp tiếp theo là tạo bộ kiểm tra gói tin có ACK trùng lặp để đếm số lượng gói tin ACK được truyền lại trùng lặp tại vSwitch (hoặc vBridge), khi số lượng đạt đến ngưỡng tối đa mà được cài đặt sẵn tại vSwitch (hoặc vBridge), bộ đếm sẽ tạo gói tin RST-ACK để buộc dừng hoàn toàn kết nối. Giải pháp này hoàn toàn có thể thực hiện được bởi quá trình phân tích các gói tin khi được gửi đến vSwitch (hoặc vBridge) đều thực hiện được một cách dễ dàng trước khi gói tin được chuyển đến xử lý tại máy chủ ảo. Từ đó, hạn chế được tối đã việc giả mạo của kẻ tấn công vào máy chủ ảo thông qua các cơ chế giao tiếp đơn giản của các máy VM hiện nay.

C. Vấn đề đạo đức

Trong thực nghiệm tấn công trên máy ảo và Docker, chúng tôi đã ngắt kết nối Internet khi tiến hành các thí nghiệm và chỉ thử nghiệm các kỹ thuật tấn công với các máy tính trong phòng thí nghiệm trong quá trình này. Các máy tính này chỉ kết nối với nhau trong quá trình thử nghiệm và không kết nối với mạng cục bộ của trường đại học của chúng tôi. Do đó, các thí nghiệm của chúng tôi hoàn toàn vô hại với Internet.

VI. KẾT LUẬN

Đối với các phần mềm phổ biến được xây dựng bởi các bên chuyên nghiệp, chẳng hạn như Nginx, kỹ thuật tấn công này rất khó thực hiện. Tuy nhiên, ý nghĩa của nó đối với hệ thống máy chủ ảo vẫn rất rõ nét vì một khi nó được thực hiện thành công, nó gây ảnh hưởng nhiều tới máy chủ ảo. Bài báo này nhấn mạnh việc cần quan tâm đúng đắn vào xử lý các lỗ hổng về "timeout" với trạng thái CLOSE-WAIT trong kết nối TCP khi xây dựng và phát triển phần mềm mới có sử dụng kết nối TCP. Lỗ hổng này có thể dẫn đến một cuộc tấn công từ chối dịch vụ dựa trên bão gói tin ACK không giới hạn về thời gian có thể gây hại cho máy chủ. Cuối cùng, TCP vẫn là một giao thức quan trọng và được sử dụng trong nhiều phần mềm ứng dụng, nó đã được xây dựng từ lâu nên chắc chắn sẽ có lỗ hổng, vì vậy việc tìm ra các lỗ hổng và nghiên cứu kỹ các lỗ hổng nhằm tìm ra các biện pháp để ngăn chặn và khắc phục các lỗ hổng này là công việc cần thiết của tất cả các chuyên gia bảo đảm an toàn thông tin. Trong bài báo này, chúng tôi đã tập trung vào nghiên cứu và phân tích sâu sắc các mô hình tấn công từ chối dịch vụ dựa trên bão gói tin ACK của tác giả Abramov [3] và tác giả Sơn [5] cũng như đề xuất một phiên bản tấn công mới để chiếm các cổng mạng của máy chủ ảo. Trong tương lai, chúng tôi sẽ tập trung nghiên cứu các hệ thống máy chủ ảo hóa khác như Hyper-V, Oracle, v.v ... với các loại tấn công này cũng như đề xuất các biện pháp tấn công mới và các biện pháp đối phó hiệu quả nhằm hạn chế các rủi ro tấn công mạng cho các ứng dụng thực tế.

TÀI LIỆU THAM KHẢO

[1] *The 2020 State of Virtualization Technology*. (Spiceworks reports, 2019), https://www.spiceworks.com/marketing/reports/state-of-virtualization. (Last accessed 18 April 2020)

[2] Imperva's DDoS Attack reports. [Online], Available at: https://www.imperva.com/blog/this-ddos-attack-unleashed-the-most-packets-per-second-ever-heres-why-thats-important. (Last accessed 18 April 2020) https://www.imperva.com/blog/2019-global-ddos-threat-landscape-report. (Last accessed 18 April 2020)

[3] R. Abramov and A. Herzberg, "TCP ack storm DoS attacks", Proceedings of the 26th IFIP TC 11 International Information Security Conference (SEC 2011), pp.12–27, June 2011.

[4] Tran Nam K., Nguyen Kim T., Thanh T.M., "An Attempt to Perform TCP ACK Storm Based DoS Attack on Virtual and Docker Network," In: Vo NS., Hoang VP. (eds) Industrial Networks and Intelligent Systems (INISCOM), vol 334, 2020.

[5] N.D. Son, M. Mimura and H. Tanaka, "An Analysis of TCP ACK Storm DoS Attack on Virtual Network", 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, pp. 288-293, 2019. (Last accessed 20 May 2020)

[6] Wireshark: Network protocol analyzer, Wireshark v3.2.3. [Online], Available at: https://wireshark.org/docs/man-pages/wireshark.html. (Last accessed 20 May 2020)

[7] J. McTigue, *2013 Virtualization Management Survey*. (InformationWeek Reports, November 2012).

[8] VMWare: Workstation for Windows, VMWare Workstation Pro 14.[Online], Available at: https://www.vmware.com/products/workstation. (Last accessed 10 April 2020)

[9] Docker: Docker desktop for Windows, Docker Nginx. [Online], Available at: https://www.docker.com/. (Last accessed 12 April 2020)

[10] A. Bansal and P. Goel, "Simulation and Analysis of Network Address Translation (NAT) & Port Address Translation (PAT) Techniques," in Int. Journal of Engineering Research and Application, ISSN : 2248-9622, Vol. 7, Issue 7, (Part 2) July 2017, pp.50-56.

[11] RFC 793 TRANSMISSION CONTROL PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION, pp.72–73, Sep 1981.

[12] Bays, L.R.; Oliveira, R.R.; Barcellos, M.P., "Gaspary, L.P.; Madeira, E.R. Virtual network security: Threats, countermeasures, and challenges," J. Internet Serv. Appl. vol. 6, issue 1, 2015.

[13] Nathiya T., Suseendran G., "An Effective Hybrid Intrusion Detection System for Use in Security Monitoring in the Virtual Network Layer of Cloud Computing Technology," In Data Management, Analytics and Innovation, Advances in Intelligent Systems and Computing, Balas, V., Sharma, N., Chakrabarti, A., Eds.; Springer: Singapore, Volume 839, 2019.

[14] Pan W., Zhang Y., Yu M., Jing J., "Improving virtualization security by splitting hypervisor into smaller components," In IFIP Annual Conference on Data and Applications Security and Privacy, Volume 7371, pp. 298–313, 2012.

[15] Mthunzi S.N., Benkhelifa E., Alsmirat M.A. Jararweh Y., "Analysis of VM communication for VM-based cloud security systems," In Proceedings of the 2018 Fifth International Conference on Software Defined Systems (SDS), pp. 182–188, 2018.

[16] Elmrabet Z., Elghazi H., Sadiki T., Elghazi H., "A New Secure Network Architecture to Increase Security among Virtual Machines in Cloud Computing," In Advances in Ubiquitous Networking, Volume 366, 2016.

[17] Sathya Narayana K., Pasupuleti S.K., "Trusted Model for Virtual Machine Security in Cloud Computing," In Progress in Computing, Analytics and Networking, Volume 710, 2018.

[18] J. Chelladhurai, P.R. Chelliah and S. Kumar, "Securing Docker Containers from Denial of Service (DoS) Attacks," 2016 IEEE International Conference on Services Computing (SCC), San Francisco, CA, pp. 856–859, 2016.

[19] A. Blenk, A. Basta, M. Reisslein and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," IEEE Communications Surveys and Tutorials, 2016, pp.655–685.

[20] E. Bauman, G. Ayoade and Z. Lin., "A Survey on Hypervisor-Based Monitoring: Approaches, Applications, and Evolutions." ACM Computing Surveys, pp.1–33, 2015.

[21] N.M. Chowdhury, K. Mosharaf and R. Boutaba, "A survey of network virtualization", Computer Networks, pp.862–876, 2010.

[22] A. Fischer, J.F. Botero, M.T Beck, H. de Meer and X. Hesselbach, "Virtual Network Embedding: A Survey," IEEE Communications Surveys & Tutorials, 2013.

[23] Arch Linux: Network bridge. https://wiki.archlinux.org/index.php/Network_bridge (Last accessed 30 June 2020)

[24] V. Nuutti, "Anatomy of a Linux Bridge," Proceedings of Seminar on Network Protocols in Operating Systems, pp.58, 2012.

[25] T. Darshan, A. Akshai C. Nirbhay, "Virtualization vulnerabilities, security issues, and solutions: a critical study and comparison," International Journal of Information Technology, 2019.

[26] A. Mallika, A. Ahsanb, M. Md. Z. Shahadata J.C. Tsouc, "Man-in-the-middle-attack: Understanding in simple words," International Journal of Data and Network Science, 2019.

[27] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J.A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelink P. Zimmermann, "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications, 2015.

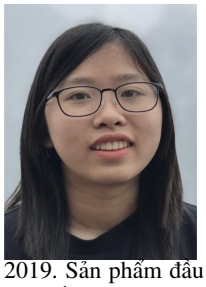
[28] K.M. Jain, M.V. Jain J.L. Borade, "A Survey on Man in the Middle Attack," International Journal of Science Technology Engineering , 2016.

[29] R.L. Bull, J.N. Matthews K. A. Trumbull, "VLAN hopping, ARP poisoning and Man-In-The-Middle Attacks in Virtualized Environments," DEF CON 24, 2016.

PROPOSE TO PERFORM NETWORK ATTACK ON VIRTUAL NETWORK USING ACK-STORM AND DEFENSE SOLUTION CONSIDERING

***Tóm tắt*—Recently, the server virtualization (hypervisor) market is growing up fast because server virtualization has many benefits. More and more businesses use hypervisors as an alternative solution to a physical server. However, hypervisors are more vulnerable than traditional servers according to recent researches. Therefore, stand on the position of a system administrator, it's necessary to prepare for the worst circumstances, understand clearly, and research for new threats that can break down the virtual system. In this paper, we attempt to perform TCP ACK storm based DoS (Denial of Service) attack on virtual and Docker networks and propose some solutions to prevent them.**

***Từ khóa*—DoS, Hypervisor, TCP, ACK storm, Virtual network, Docker network.**



Phùng Vũ Hạnh My đang học trường THPT Chuyên Đại học Sư phạm. Lĩnh vực quan tâm của Hạnh My là Toán và Khoa học máy tính. My là học viên của Trung tâm MindX từ 2019, học các khóa về thiết kế web, ngôn ngữ lập trình, design; Đạt Huy chương Đồng giải team tại kỳ thi International Mathermatics Competiton tại Nam Phi 2019. Sản phẩm đầu tay game Pandemic Choice đã lọt vòng bán kết cuộc thi AppJamming 2021. My đang tham gia trại hè online về AI do Inspirit AI (Mỹ) tổ chức và đăng ký tham gia trại hè MaSSP 2021 về lĩnh vực Data Science.

Trần Nam Khánh nhận bằng kỹ sư CNTT tại trường Đại học Lê Quý Đôn, Hà Nội vào năm 2020. Ông Khánh là giảng viên của trường Đại học Lê Quý Đôn từ năm 2021. Lĩnh vực nghiên cứu nằm trong các lĩnh vực an ninh mạng, công nghệ mạng, xử lý ảnh.

Tạ Minh Thanh nhận bằng kỹ sư CNTT và Thạc sĩ Khoa học Máy tính của Học viện Phòng vệ Nhật Bản, vào năm 2005 và 2008. Ông Thanh là giảng viên của trường Đại học Lê Quý Đôn từ năm 2005. Năm 2015, ông nhận bằng Tiến sĩ Khoa học Máy tính của Học viện Công nghệ Tokyo, Nhật Bản. Ông đã được công nhận của Hội đồng Giáo sư nhà nước chức danh Phó giáo sư vào năm 2019. Ông cũng là thành viên của Hiệp hội IPSJ Nhật Bản và Hiệp hội IEEE. Lĩnh vực nghiên cứu của ông nằm trong lĩnh vực thủy vân, an ninh mạng và thị giác máy tính.

Nguyễn Kim Thanh nhận bằng kỹ sư CNTT và Thạc sĩ Công nghệ mạng của Đại học Kỹ thuật Điện tử Tây An, Trung Quốc vào năm 2012 và 2018. Ông Thanh là giảng viên của trường Đại học Lê Quý Đôn từ năm 2013. Lĩnh vực nghiên cứu của ông nằm trong lĩnh vực Công nghệ mạng, Điện toán đám mây, An ninh mạng.