NGHIÊN CỨU VỀ AN NINH CÔNG NGHỆ MẠNG ĐỊNH NGHĨA BẰNG PHẦN MỀM SDN VÀ ỨNG DỤNG

Nguyễn Thị Thảo*, Trần Vũ Hà, Lương Minh Quân

Khoa Công nghệ thông tin, Học viện Nông nghiệp Việt Nam

**Tác giả liên hệ: ntthao81@vnua.edu.vn*

Ngày nhận bài: 02.08.2021 Ngày chấp nhận đăng: 15.08.2022

TÓM TẮT

Trong bài báo này chúng tôi tập trung nghiên cứu công nghệ mạng định nghĩa bằng phần mềm (Software Defined Network - SDN) và khả năng ứng dụng của nó. SDN được xây dựng dựa trên một kiến trúc linh hoạt, dễ quản lý, hiệu suất cao và thích nghi tốt. Công nghệ này lý tưởng cho các ứng dụng đòi hỏi băng thông cao và cần sự linh hoạt. Tuy nhiên, nó cũng đặt ra những thách thức trong việc đảm bảo an ninh và an toàn dữ liệu. Để hiểu rõ hơn về SDN, chúng tôi nghiên cứu các đặc điểm, ưu điểm, nhược điểm của mạng, sau đó so sánh mạng SDN với mạng truyền thống. Tiếp theo, chúng tôi thảo luận về các mối đe dọa bảo mật mà mạng SDN gặp phải và các kỹ thuật có thể sử dụng để ngăn chặn và giảm thiểu rủi ro cho mạng. Để xem xét khả năng ứng dụng trong thực tế, nhóm đã cài đặt thử nghiệm xây dựng mạng SDN trên trên phần mềm Mininet và Onos để có cái nhìn tổng quát hơn về công nghệ mạng SDN và các lỗ hổng trong bảo mật trong mạng.

Từ khóa: Mạng định nghĩa bằng phần mềm, SDN.

Research on Security of Software-definednetworking (SDN) and Application

ABSTRACT

In this article, we focus on software-defined networking (SDN) technology. SDN is a flexible, manageable, high-performance, and adaptable architecture. SDN has shown its potentials for application that require high bandwidth and flexibility. However, it also poses challenges in terms of data security. To better understand SDN, we study the characteristics, advantages and disadvantages of SDN, compare it with traditional network. Next, we discuss the security threats in a SDN-based network. We also present some techniques that can be used to prevent and mitigate the risks to the network. Along with theory about SDN, we implement a demo SDN system using Mininet and Onos. This virtual system is simple but it shows all the advantage and disadvantage of a typical software-defined network.

Keywords: Software Defined Network, SDN.

1. ĐẶT VẤN ĐỀ

Mạng Internet ra đời đã tạo nên một cuộc cách mạng trong công nghệ thông tin. Nó giúp mọi sự giao tiếp và trao đổi kiến thức, thông tin của con người trở nên dễ dàng hơn tạo nền tảng cho nền kinh tế tri thức hiện nay. Tuy nhiên, kiến trúc mạng truyền thống đã không hề có sự thay đổi trong nhiều năm qua và đang ngày càng trở nên không phù hợp với nhu cầu kinh doanh của các doanh nghiệp, các nhà khai thác mạng cũng như người dùng cuối. Hiện nay nhu cầu về nghiệp vụ ngày càng phức tạp của các doanh nghiệp và mức độ đa dạng về ứng dụng đang ngày càng gia tăng, nhu cầu khác nhau của người dùng về mạng kết nối. Mạng cần phải đáp ứng việc thay đổi nhanh chóng các thông số về độ trễ, băng thông, định tuyến, bảo mật,… theo yêu cầu của các ứng dụng.

Hệ thống mạng truyền thống còn phức tạp, chưa đồng nhất, khó mở rộng, còn phụ thuộc nhiều vào các nhà cung cấp thiết bị, chi phí cao cùng với nhiều nguy cơ về an toàn trong bảo mật. Các khó khăn và phức tạp trong việc tích

hợp các giải pháp bảo mật hệ thống mạng là một vấn đề quan trọng hàng đầu. Trước những bất cập còn tồn tại hiện nay thì công nghệ mạng định nghĩa bằng phần mềm được đưa ra bởi Cisco Inc, White paper (2013) (SDN - Software Defined Network) là một giải pháp cho công nghệ mạng hiện nay.

Công nghệ SDN là kiến trúc linh hoạt, dễ quản lý, hiệu suất cao và thích nghi tốt, khiến công nghệ này lý tưởng cho các ứng dụng đòi hỏi băng thông cao và cần sự linh hoạt hiện nay, đồng thời nó cũng đặt ra những thách thức trong việc đảm bảo an ninh và an toàn dữ liệu trong tương lai.

Công nghệ SDN là công nghệ còn khá mới hiện nay nhưng đã được nhiều công ty và tập đoàn lớn chú trọng phát triển, từ đó ta thấy SDN là một tầm nhìn của kiến trúc mạng trong tương lai, với giao thức OpenFlow là thành phần chính và là cốt lõi của việc phát triển mạng SDN.

Để áp dụng được linh hoạt, hiệu quả công nghệ mạng mới trên, cần hiểu được cấu trúc cũng như hoạt động của giao thức OpenFlow và hệ thống mạng SDN, từ đó đưa ra giải pháp thiết kế vận hành cũng như khắc phục lỗi, các lỗ hổng trong hệ thống mạng SDN; đưa ra được lợi ích, ưu nhược điểm của hệ thống mạng SDN, những thuận lợi và khó khăn khi triển khai mạng SDN. Từ đó đưa ra được các điểm yếu, điểm nhạy cảm dễ bị tấn công trong mạng SDN. Thấy được mối nguy hiểm trong mạng và cách thức tấn công của hacker từ đó đưa ra giải pháp phòng chống trong mạng.

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Công nghệ mạng SDN

Hiện nay có rất nhiều định nghĩa về mạng SDN nhưng theo ONF (Open Networking Foundation - một tổ chức phi lợi nhuận đang hỗ trợ việc phát triển SDN thông qua việc nghiên cứu các tiêu chuẩn mở phù hợp) thì mạng SDN được định nghĩa như sau: "Mạng định nghĩa bằng phần mềm hay Sofware Defined Network (SDN) là một kiểu kiến trúc mạng mới, năng động, dễ quản lý, chi phí hiệu quả, dễ thích nghi và rất phù hợp với nhu cầu mạng ngày càng tăng hiện nay. Kiến trúc này phân tách phần điều khiển mạng (Control Plane) và chức năng vận chuyển dữ liệu (Forwarding Plane hay Data Plane), điều này cho phép việc điều khiển mạng có thể lập trình được dễ dàng và cơ sở hạ tầng mạng độc lập với các ứng dụng và dịch vụ mạng".

2.1.1. Ý tưởng mạng SDN

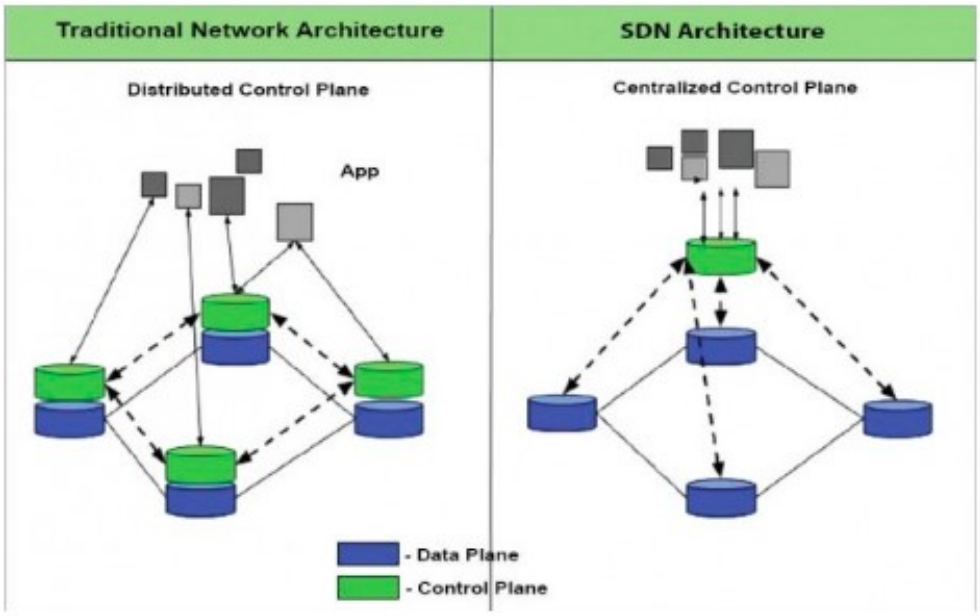
Trong mạng truyền thống, các thiết bị mạng đóng cả vai trò logic trong định tuyến, điều khiển hoạt động của mạng lẫn chuyển tiếp stream các gói tin, dữ liệu. Mỗi thiết bị đều có cả hai chức năng này. Do đó, khi cấu hình một hệ thống mạng nhiều thành phần, quản trị viên phải cấu hình thủ công từng thiết bị bằng tay và các thiết bị không hề có sự liên hệ với nhau về mặt quản trị. Do đó việc cấu hình, vận hành trở nên đặc biệt khó khăn, các thiết bị hoàn toàn độc lập do đó chỉ có thể thay đổi cấu hình trên từng thiết bị thủ công một cách tuần tự.

Công nghệ mạng được định nghĩa bằng phần mềm (Software-defined network - SDN) là một cách tiếp cận theo hướng điện toán đám mây, tức là tập trung hóa việc quản trị thiết bị mạng, giảm thiểu công việc riêng lẻ, tốn nhiều thời gian trên từng thiết bị cụ thể, tạo điều kiện thuận lợi cho việc quản lý mạng và cho phép cấu hình mạng hiệu quả bằng các chương trình được lập trình để cải thiện hiệu suất và tăng cường khả năng giám sát mạng. SDN nhằm mục tiêu giải quyết vấn đề kiến trúc tĩnh, phi tập trung, phức tạp của các mạng truyền thống không phù hợp với yêu cầu về tính linh hoạt và xử lý sự cố dễ dàng của các hệ thống mạng hiện tại (Hình 1).

Có thể thấy sự khác biệt cơ bản giữa mạng truyền thống và mạng SDN là:

- Phần điều khiển và phần vận chuyển dữ liệu trên mạng truyền thống đều được tích hợp trong thiết bị mạng trong khi mạng SDN, phần điều khiển được tách riêng khỏi thiết bị mạng và được chuyển đến một thiết bị được gọi là bộ điều khiển SDN.

- Phần thu thập và xử lý các thông tin: Đối với mạng truyền thống, phần này được thực hiện ở tất cả các phần tử trong mạng còn trong mạng SDN, phần này được tập trung xử lý ở bộ điều khiển SDN.



Ghi chú: Traditional Network Architecture: Kiến trúc mạng truyền thống; Distributed Control Plane: điều khiển phân tán; SDN Architecture: Kiến trúc mạng SDN; Centralized Control Plane: điều khiển tập trung; Control Plane: phần điều khiển; Data Plane: phần vận chuyển dữ liệu.

Hình 1. So sánh kiến trúc mạng truyền thống và mạng SDN

- Mạng truyền thống không thể được lập trình bởi các ứng dụng. Các thiết bị mạng phải được cấu hình một cách riêng lẻ và thủ công. Trong khi đối với mạng SDN, mạng có thể lập trình bởi các ứng dụng, bộ điều khiển SDN có thể tương tác đến tất cả các thiết bị trong mạng.

Phần điều khiển được tách rời và được tập trung ở bộ điều khiển SDN. Điều này có nghĩa là các thiết bị mạng ở lớp thiết bị phần cứng không cần phải hiểu và xử lý các giao thức phức tạp mà chúng chỉ nhận và vận chuyển dữ liệu theo một đường nào đó dưới sự chỉ huy của bộ điều khiển SDN. Dựa vào bộ điều khiển SDN mà các nhà khai thác và quản trị mạng có thể lập trình cấu hình trên đó thay vì phải thực hiện thủ công hàng ngàn câu lệnh cấu hình trên các thiết bị riêng lẻ. Điều này giúp triển khai các ứng dụng mới và các dịch vụ mạng một cách nhanh chóng.

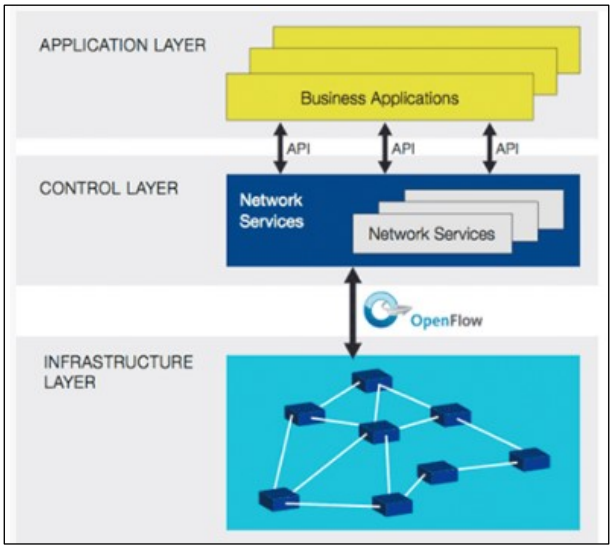
2.1.2. Kiến trúc mạng SDN

Về cơ bản kiến trúc của SDN bao gồm ba lớp: lớp ứng dụng (Appication Layer), lớp điều khiển (Control Layer) và lớp hạ tầng (Infrastructure Layer).

Lớp ứng dụng chứa các ứng dụng hoặc các chức năng mạng điển hình mà các tổ chức sử dụng như các hệ thống phát hiện xâm nhập, cân bằng tải hoặc tường lửa. Điều này khác với một mạng truyền thống, với kiến trúc mạng truyền thống, để thực hiện các chức năng này, mạng sẽ cần sử dụng một thiết bị chuyên dụng, như tường lửa hoặc thiết bị cân bằng tải riêng biệt. Trong khi với SDN các thiết bị này sẽ được thay thế bằng một ứng dụng phần mềm sử dụng bộ điều khiển để quản lý xử lý dữ liệu.

Lớp điều khiển đại diện cho phần mềm điều khiển SDN tập trung. Nó hoạt động như bộ não của mạng. Bộ điều khiển nằm trên một máy chủ và quản lý các chính sách và luồng lưu lượng trên toàn mạng.

Lớp hạ tầng bao gồm các thiết bị chuyển mạch vật lý trong mạng. Thiết bị chuyển mạch nhận chỉ thị và quy tắc định hướng khi cần thiết từ bộ điều khiển. Các chuyển mạch này sẽ cung cấp cho bộ điều khiển thông tin về lưu lượng mà nó xử lý.



Ghi chú: Appication Layer: Lớp ứng dụng; Control Layer: Lớp điều khiển; Infrastructure Layer: Lớp hạ tầng.

Hình 2. Kiến trúc phân tầng mạng SDN

Khi nhắc đến SDN, chúng ta cần đề cập đến OpenFlow, thực chất thì OpenFlow chính là một giao thức chính trong SDN, cho phép giao tiếp giữa lớp hạ tầng và lớp điều khiển. Trên thực tế có rất nhiều các giao thức khác tồn tại, nhưng OpenFlow vẫn tương đối phổ biến vì lí do lịch sử và là tiêu chuẩn đầu tiên cho các giao diện lập trình ứng dụng API cầu nam (southbound API) (Hình 2).

Như hình 2 đã chỉ ra, mạng SDN gồm ba tầng, ba tầng này giao tiếp với nhau bằng cách sử dụng các giao diện lập trình ứng dụng API cầu bắc (northbound API) và cầu nam (southbound API). Các phần mềm tầng ứng dụng giao tiếp với lớp điều khiển thông qua northbound API, lớp điều khiển và các thiết bị chuyển mạch (thuộc lớp hạ tầng) giao tiếp thông qua southbound API. Southbound API sử dụng giao thức OpenFlow, northbound API chưa có tiêu chuẩn chung.

2.1.3. Ưu điểm mạng SDN

Khả năng lập trình cho mạng: SDN cho phép kiểm soát hành vi mạng bằng phần mềm nằm ngoài các thiết bị cung cấp kết nối vật lý của mạng. Do đó, các nhà khai thác mạng có thể lập trình, điều chỉnh hành vi của các mạng để hỗ trợ các dịch vụ mới và hỗ trợ khách hàng cá nhân một cách dễ dàng (Sama & cs., 2015). Bằng cách tách phần cứng khỏi phần mềm, các nhà khai thác có thể giới thiệu các dịch vụ mới nhanh chóng, không bị ràng buộc bởi các nền tảng đóng và độc quyền của nhà cung cấp thiết bị mạng.

Tập trung vào sự kiểm soát thông minh: SDN được xây dựng trên các cấu trúc liên kết mạng tập trung logic, cho phép kiểm soát và quản lý thông minh tài nguyên mạng. Phương pháp điều khiển mạng truyền thống là phương pháp phân tán trong đó các thiết bị hoạt động độc lập với nhận thức hạn chế về trạng thái chung của toàn mạng. Do đó, mỗi thiết bị sẽ không thể tối ưu được hiệu năng dựa trên trạng thái chung mà chỉ có thể hoạt động riêng biệt dẫn tới khả năng làm giảm hiệu năng chung của toàn mạng. SDN cung cấp khả năng điều khiển tập trung, quản lý băng thông, khôi phục, bảo mật và do đó có thể xây dựng các chính sách thông minh, tối ưu và có tổ chức dựa trên trạng thái toàn diện của mạng.

Trừu tượng hóa mạng: Các dịch vụ và ứng dụng chạy trên công nghệ SDN được trừu tượng hóa từ các công nghệ và phần cứng cơ bản cung cấp kết nối vật lý từ bộ điều khiển mạng. Các ứng dụng sẽ tương tác với mạng thông qua các

API, thay vì các giao diện quản lý được kết hợp chặt chẽ với phần cứng.

Kiến trúc SDN mở ra một kỷ nguyên mới của sự cởi mở, cho phép khả năng tương tác của nhiều nhà cung cấp cũng như thúc đẩy một hệ sinh thái trung lập với nhà cung cấp. Sự cởi mở đến từ chính cách tiếp cận SDN. Các API mở hỗ trợ một loạt các ứng dụng, bao gồm cả điều phối dịch vụ đám mây, OSS/BSS và các ứng dụng được kết nối quan trọng. Ngoài ra, phần mềm thông minh có thể kiểm soát phần cứng từ nhiều nhà cung cấp với giao diện lập trình mở như OpenFlow. Cuối cùng, từ bên trong SDN, các ứng dụng và dịch vụ mạng thông minh có thể chạy trong môi trường phần mềm chung.

Một lợi thế chính của công nghệ SDN là khả năng cho các nhà khai thác mạng viết chương trình sử dụng API của SDN và cung cấp cho ứng dụng quyền kiểm soát hành vi mạng. SDN cho phép người dùng phát triển các ứng dụng nhận biết mạng, theo dõi thông minh trạng thái hoạt động của mạng và tự động điều chỉnh cấu hình mạng khi cần.

2.1.4. Nhược điểm của mạng SDN

Theo Dabbagh & cs. (2015), mạng SDN có thể gặp vấn đề về độ trễ. Mọi thiết bị được sử dụng trên mạng đều chiếm một không gian trên đó. Tốc độ tương tác giữa các thiết bị và mạng phụ thuộc vào số lượng tài nguyên ảo hóa. Điều này có thể dẫn đến độ trễ đáng kể.

Hiện tại, không có bất kỳ giao thức bảo mật tiêu chuẩn nào cho SDN. Mặc dù có một số nhà cung cấp dịch vụ bên thứ ba, nhưng vẫn còn nhiều lo ngại về bảo mật. Cần có nhiều kiến thức và chuyên môn để xử lý hệ thống SDN mới có thể ngăn chặn những cuộc tấn công lớn.

SDN không sử dụng các bộ định tuyến và thiết bị chuyển mạch thông thường. Do đó, bảo mật đi kèm với chúng thường bị hạn chế. Điều này dẫn tới mạng dễ bị tấn công hơn trước các mối đe dọa bên ngoài.

2.2. An ninh mạng SDN

Vấn đề an ninh, bảo mật không chỉ là lợi thế mà cũng là một thách thức đối với công nghệ SDN. SDN được quản lý và điều khiển thông qua bộ điều khiển trung tâm, theo ONF Solution Brief (2013). Hacker có thể tấn công vào bộ điều khiển và dễ dàng chiếm quyền điều khiển toàn mạng.

Một thách thức khác với SDN đó là mặc dù nó có tên là "được định nghĩa" nhưng nó lại thực sự không có định nghĩa nào thống nhất và tiêu chuẩn chung. Các nhà cung cấp khác nhau đề xuất các cách tiếp cận khác nhau cho SDN, từ các mô hình tập trung vào phần cứng, nền tảng ảo hóa cho đến các thiết kế mạng và phương thức không sử dụng bộ điều khiển. Có rất nhiều các thiết kế và định hướng khác nhau. Mặc dù SDN có thể làm việc với các công nghệ và quy trình này, nhưng về bản chất nó vẫn là một công nghệ riêng biệt.

2.3. Nghiên cứu các phần mềm thực nghiệm

2.3.1. Phần mềm ONOS

Phần mềm ONOS (Hệ điều hành mạng mở) là một nền tảng mã nguồn mở để phát triển các ứng dụng và giải pháp điều khiển mạng. Nó được thiết kế để đáp ứng nhu cầu của các mạng của nhà cung cấp dịch vụ, nó cũng có thể được áp dụng cho mạng trung tâm dữ liệu hoặc mạng trường học.

Phần mềm được viết bằng Java và cung cấp nền tảng ứng dụng SDN phân tán trên Apache Karaf OSGi. Hệ thống được thiết kế để hoạt động như một cụm các nút giống hệt nhau về ngăn xếp phần mềm của chúng và có thể chịu được sự cố của các nút riêng lẻ mà không gây gián đoạn khả năng kiểm soát hoạt động mạng của nó.

Trong khi ONOS chủ yếu dựa vào các giao thức và mô hình tiêu chuẩn, ví dụ: OpenFlow, NETCONF, OpenConfig, kiến trúc hệ thống của nó không bị ràng buộc trực tiếp với chúng. Thay vào đó, ONOS cung cấp tập hợp các mô hình và mô hình trừu tượng cấp cao của riêng mình, mà nó cung cấp API cho các lập trình viên ứng dụng. Các mô hình này có thể được mở rộng bởi các ứng dụng tại thời điểm chạy.

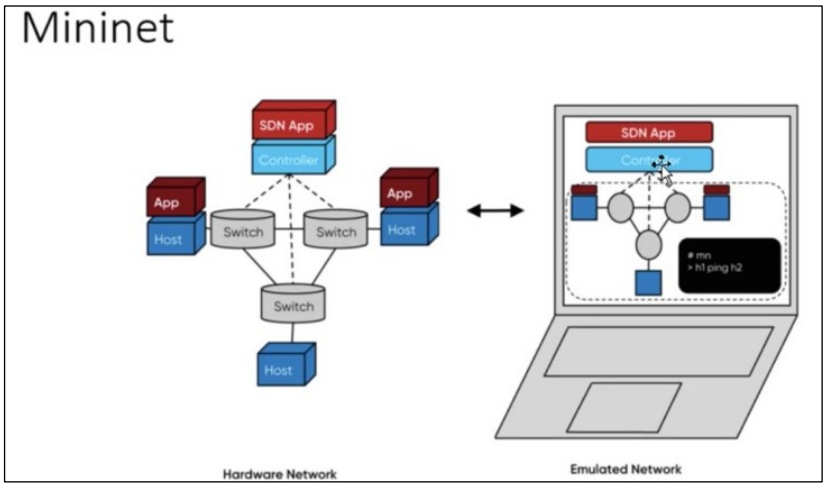
2.3.2. Phần mềm Mininet

Mininet là một công cụ giả lập mạng, bao gồm tập hợp các hosts đầu cuối, các bộ chuyển

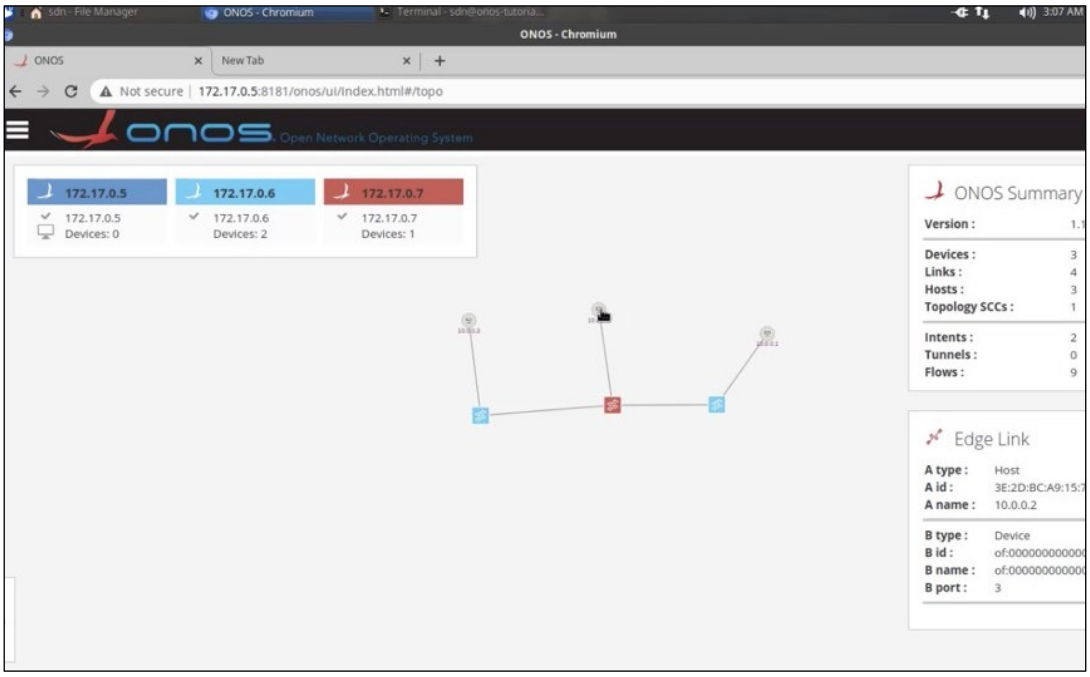
mạch (switches), các bộ định tuyến (routers) và các liên kết trên một Linux kernel. Mininet sử dụng công nghệ ảo hóa (ở mức đơn giản) để tạo nên hệ thống mạng hoàn chỉnh, chạy chung trên cùng một kernel, hệ thống và user code.

Các máy chủ (host) ảo, chuyển mạch (switch), liên kết và các controller trên Mininet là các thực thể được giả lập dưới dạng phần mềm thay vì phần cứng. Trong đó, host Mininet có thể chạy bất kì phần mềm nào đã cài trên hệ thống Linux (môi trường mà Mininet đang chạy). Các phần mềm này có thể gửi gói tin thông các ethernet interface của mininet với tốc độ liên kết và trễ đặt trước.

Mininet cho phép tạo mạng nhanh chóng, tùy chỉnh được mạng, chạy được các phần mềm thực sự như web servers, TCP monitoring, Wireshark; tùy chỉnh được việc chuyển tiếp gói tin. Mininet cũng dễ dàng sử dụng và không yêu cầu cấu hình đặc biệt gì về phần cứng để chạy: Mininet có thể cài trên laptop, server, VM, cloud (Linux).



Hình 3. Mô phỏng hệ thống bằng Mininet



Hình 4. Mạng LAN ảo SDN với 3 controller và 3 switch

3. KẾT QUẢ VÀ THẢO LUẬN

3.1. Kết quả

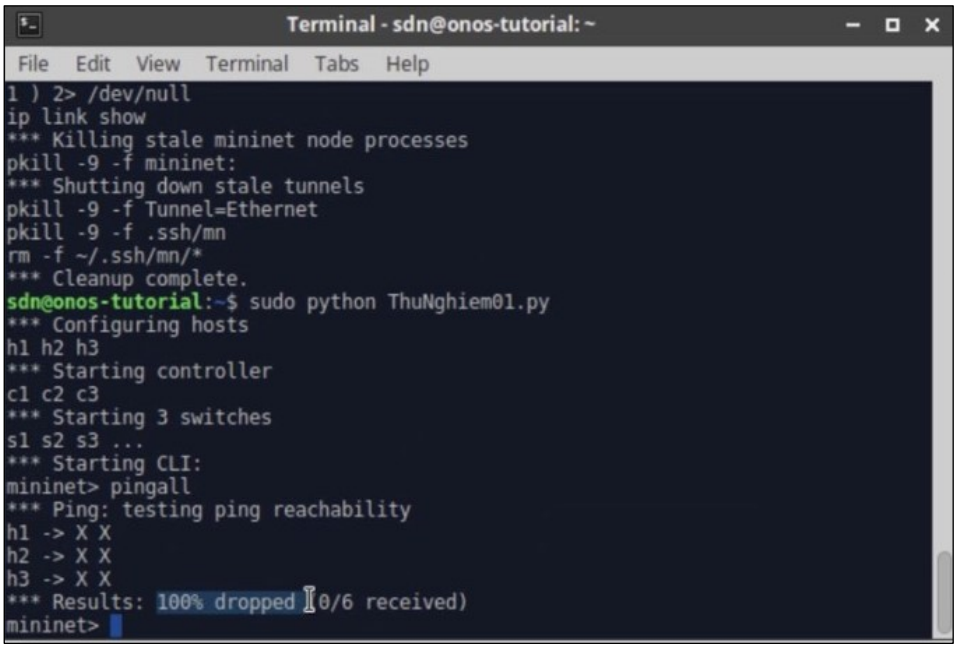
3.1.1. Thử nghiệm 01 (tạo mạng SDN ảo)

Dùng Mininet để mô phỏng các Switch và các host, các controller thì dùng mã nguồn mở ONOS để mô phỏng (Hình 3).

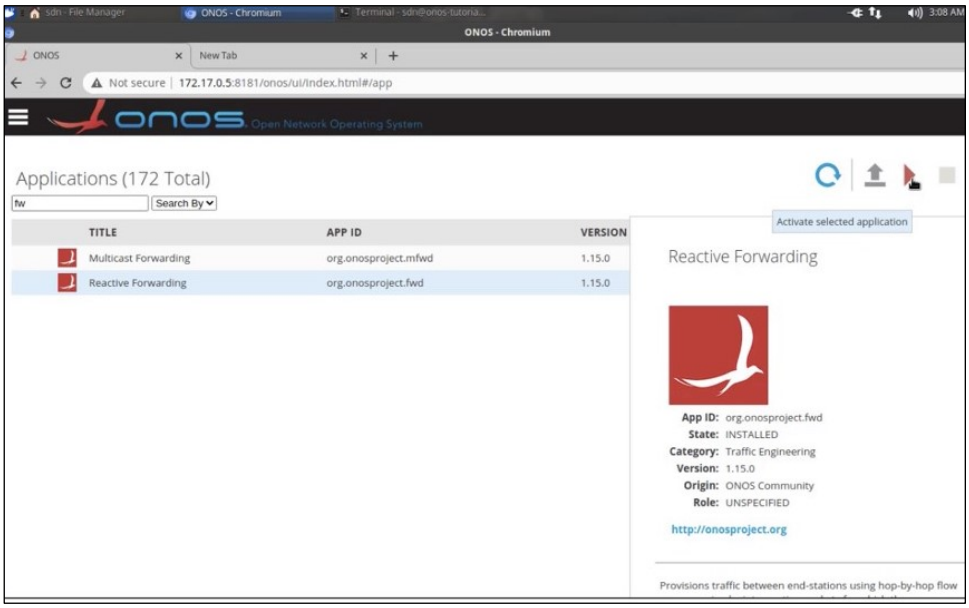
Xây dựng 3 switch kết nối với 3 controller lần lượt là: 172.17.0.5, 172.17.0.6, 172.17.0.7 (Hình 4).

Tuy nhiên, hiện tại các máy chưa thông, kiểm tra lại bằng lệnh pingall, kết quả là 100% dropped tức không gửi được các gói tin (Hình 5).

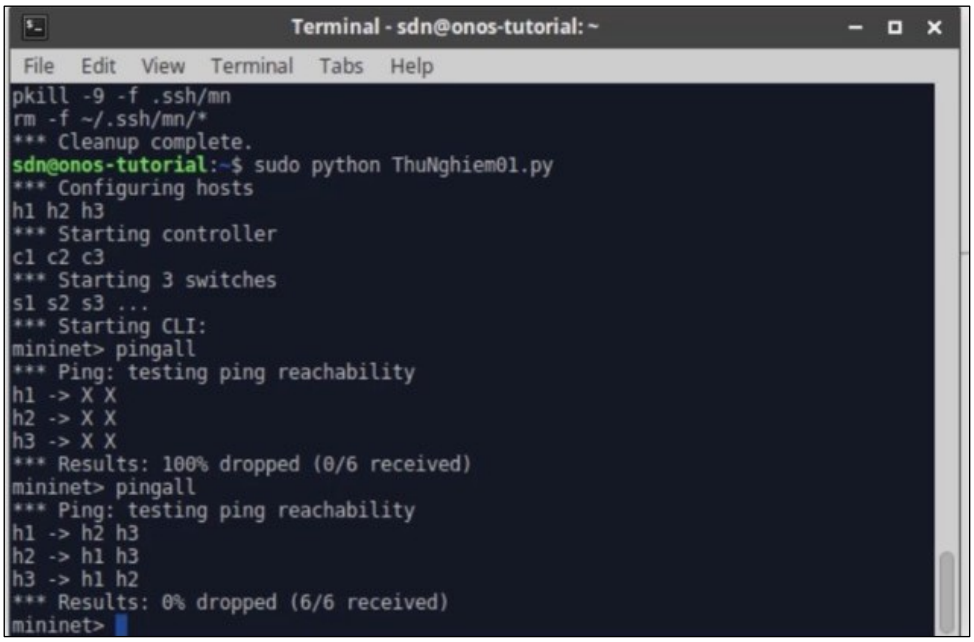
Thực hiện ý tưởng SDN, trên controller có các phần mềm để tạo các hệ thống hoạt động được, tức là các máy thông với nhau. Sử dụng Reactive Forwarding để kích hoạt và kiểm tra lại (Hình 6).



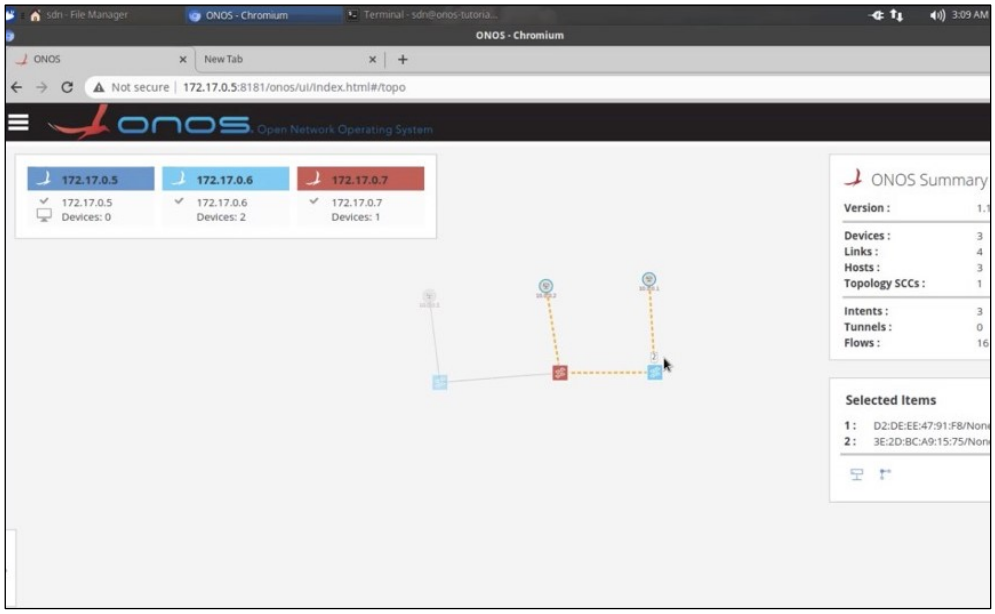
Hình 5. Kiểm tra hệ thống khi chưa thông mạng



Hình 6. Kích hoạt hệ thống bằng Reactive Forwarding



Hình 7. Kiểm tra lại hệ thống sau khi kích hoạt



Hình 8. Hình ảnh mạng sau khi các thiết bị trong mạng được kết nối

Kiểm tra lại bằng lệnh pingall, 0% dropped tức toàn bộ 6 gói tin đều được chuyển (Hình 7).

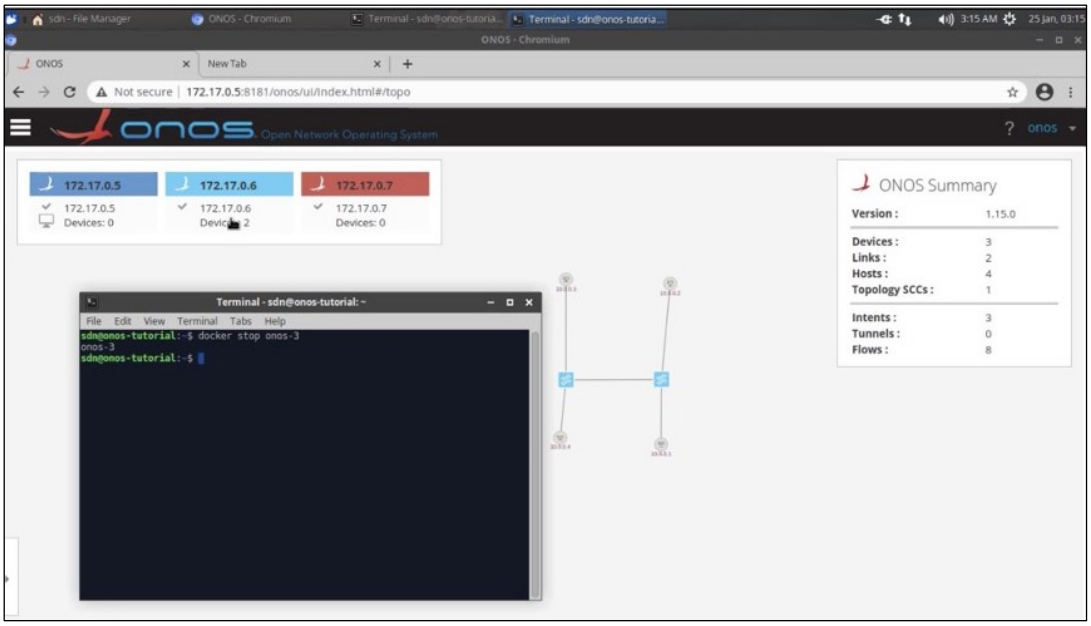
Kiểm tra lại các máy hiện tại đã được kết nối (Hình 8).

3.1.2. Thử nghiệm tạo cluster controller

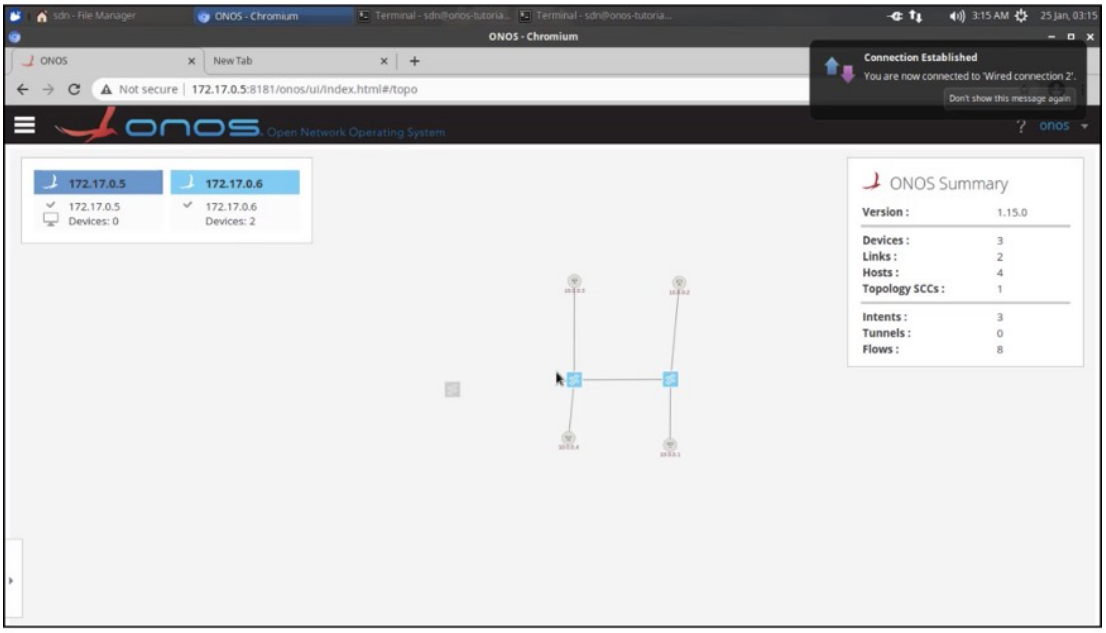
Tạo mạng LAN ảo gồm 3 controller 172.17.0.5, 172.17.0.6, 172.17.0.7, 4 host được kết nối với nhau (Hình 9).

Thử nghiệm tắt 1 controller 172.17.0.7 và kiểm tra lại hệ thống (Hình 10, 11).

Khi một controller bị hỏng hệ thống mạng vẫn hoạt động bình thường nhờ vào cụm các controller. Với mạng SDN có 1 controller điều khiển toàn mạng, nếu controller hỏng thì toàn mạng sẽ hỏng vì thế thì khi triển khai thực tế cần 1 cụm cluster là giải pháp cần thiết để đảm bảo hệ thống mạng hoạt động ổn định nhất có thể.



Hình 9. Mạng LAN SDN ảo gồm 3 controller và 4 switch



Hình 10. Hình ảnh mạng SDN ảo sau tắt 1 controller

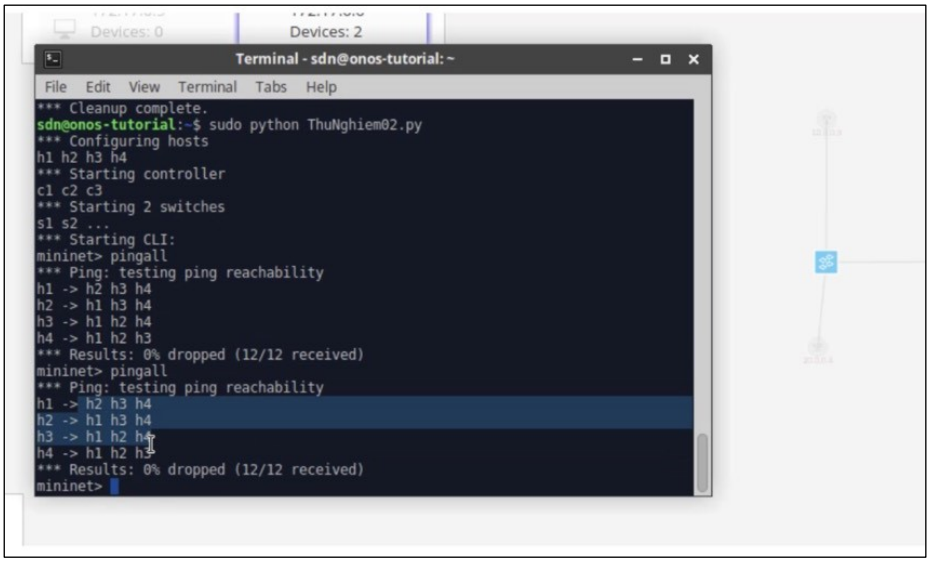
3.1.3. Thử nghiệm giám sát hệ thống mạng

Trong mạng SDN mỗi máy tính có thể đóng vai trò controller, việc giám sát controller trong mạng SDN rất quan trọng, nếu controler hỏng thì mạng sẽ có vấn đề.

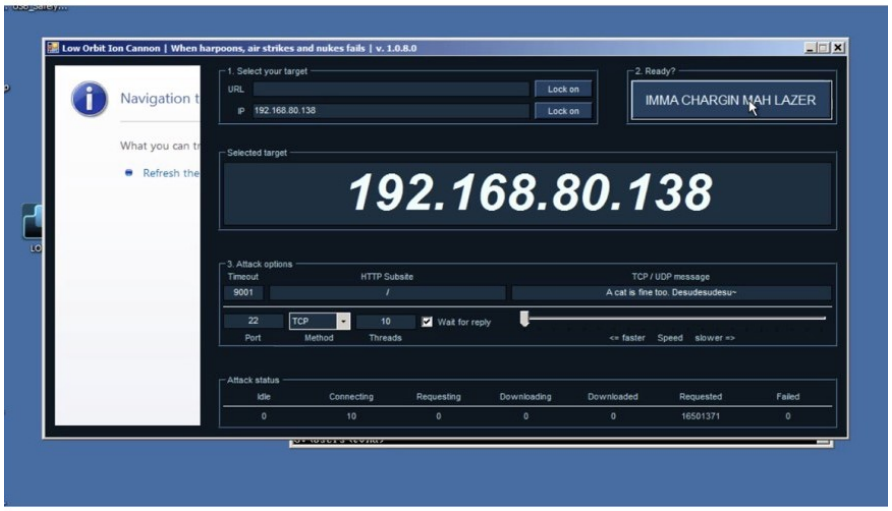
Thử nghiệm dùng tấn công vào máy tính nạn nhân 192.168.80.138 (Hình 12).

Kiểm tra Task Manager, lúc này máy nạn nhân liên tục nhận được gói tin của bên tấn công, CPU hoạt động công suất cao (Hình 13).

Thử nghiệm dừng tấn công, nhấn Stop flooding, kiểm tra lại bên máy nạn nhân, bắt gói tin nhận thấy thưa thớt hơn, rất ít gói tin từ phía tấn công, trạng thái CPU hoạt động không bị quá tải, chỉ ở mức 6-7% (Hình 14).



Hình 11. Kiểm tra thông mạng sau khi tắt 1 controller



Hình 12. Thử nghiệm tấn công vào máy tính nạn nhân

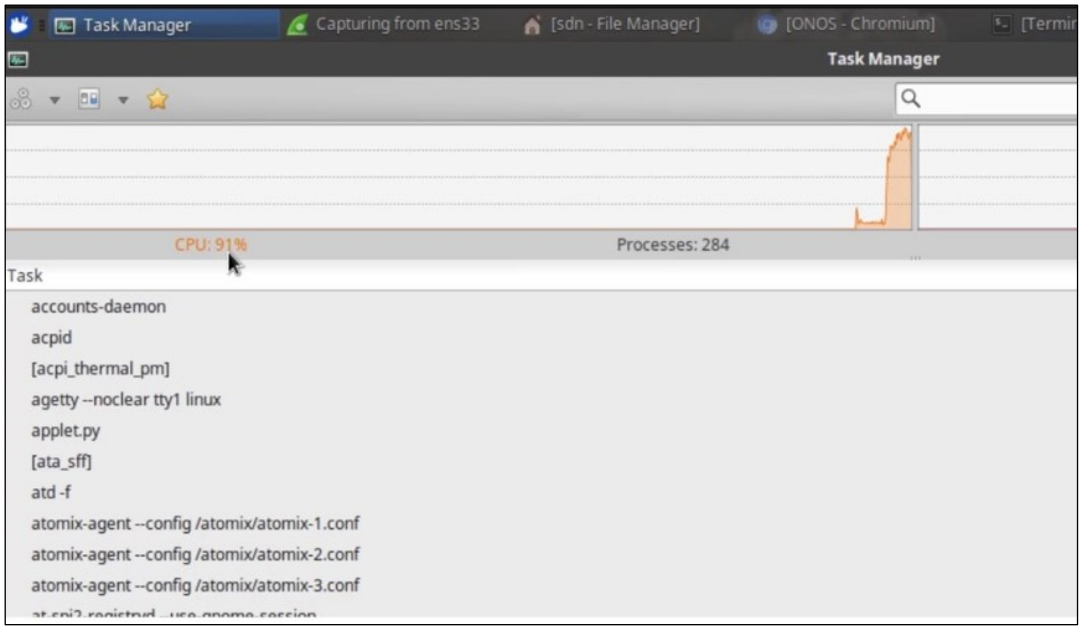
Trên đây là hai công cụ Task manager và Wireshark có sẵn trên máy tính giúp chúng ta có thể kiểm tra giám sát hoạt động của máy tính, qua đó có thể chuẩn đoán cũng như xử lý thích hợp khi hệ thống mạng có vấn đề.

3.2. Thảo luận

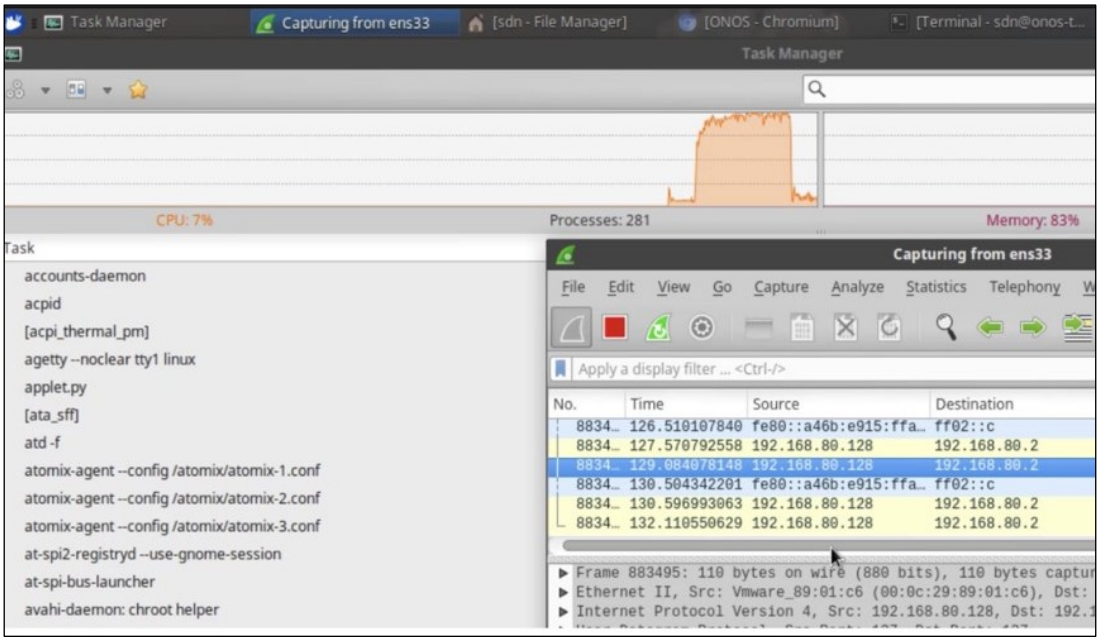
Qua các thử nghiệm được xây dựng trên trên phần mềm Mininet và Onos, nhóm nghiên cứu đã có thêm cái nhìn tổng quát hơn về công nghệ mạng SDN và các lỗ hổng trong bảo mật.

Tấn công DoS ở đây được dùng chung cho các kiểu tấn công từ chối dịch vụ như DoS, DDoS, DrDos... Tấn công DoS làm cho thành phần điều khiển quá tải trong xử lý gói tin mới, chiếm dụng băng thông, gây tràn và ngập lụt đường truyền. Kẻ tấn công gửi liên tiếp các gói tin yêu cầu thiết lập kết nối giả mạo trong thời gian khả dụng. Trong thời gian ngắn, toàn bộ tài nguyên hệ thống bị chiếm dụng để xử lý gói tin giả mạo, gây tê liệt hệ thống.

Dựa trên những nghiên cứu này, nhóm tiến hành xây dựng các nguyên tắc thiết kế mạng SDN và kết hợp với các giải pháp bảo mật toàn diện nhằm làm cho mạng SDN bảo mật tốt hơn, an toàn hơn.



Hình 13. CPU hoạt động công suất cao khi bị tấn công



Hình 14. Trạng thái máy nạn nhân sau khi hết bị tấn công

Trong đó, các nguyên tắc, giải pháp thiết kế:

- Nâng cao khả năng chịu lỗi của toàn hệ thống bằng cách sử dụng nhiều Controller (nói cách khác là xây dựng một cluster của các controller).

- Đặt thêm các tường lửa chuyên dụng cho các máy controller hoặc server có kết nối đến Internet.

Các giải pháp bảo mật, nhóm dùng SDN controller để:

- Thiết lập quyền truy cập tới từng thiết bị cụ thể. Ví dụ trong hệ thống mạng của Khoa Công nghệ thông tin có một máy in đặt tại văn phòng khoa, người quản trị có thể thiết lập để chỉ các máy tính của nhân viên trong văn phòng hoặc của ban chủ nhiệm khoa có thể truy cập tới

máy in này. Việc kiểm soát truy cập có thể áp dụng với các server khác như file server, email server, web server...

- Thiết lập các quy tắc (rule) để kiểm soát quá trình truy cập Internet của các máy trong mạng nội bộ. Hoặc theo chiều ngược lại: quản lý truy cập của các máy từ Internet vào các mạng trong mạng cục bộ, đặc biệt là các server cục bộ.

SDN ngoài kế thừa phương pháp bảo mật trong mạng truyền thống như xây dựng cơ chế phòng chống như tường lửa Firewall, IPsec, TLS... còn bổ sung thêm phương pháp như xây dựng cơ chế dự phòng trên Controller, nguyên tắc trong xây dựng hệ thống mạng và cơ chế phục hồi.

Công nghệ mạng SDN sử dụng giao thức OpenFlow đã đang phát triển nhanh chóng hiện nay. Với những ưu thế vượt trội hơn so với công nghệ mạng truyền thống về tốc độ, khả năng mở rộng của mạng SDN giúp cho quản trị hệ thống trở nên dễ quản lý hơn. Chính vì vậy, công nghệ này đang được các nhà phát triển đón nhận tích cực và hứa hẹn sẽ là một công nghệ được sử dụng vượt trội trong tương lai gần. Qua nghiên cứu khi triển khai thực tế, nhóm tác giả đã nhận thấy có những ưu điểm và nhược điểm mạng SDN như sau:

Ưu điểm và nhược điểm:

Ưu điểm:

- Quản trị tập trung: Người quản trị không cần phải tới từng thiết bị hoặc từng server để quản lý và cấu hình như cách quản lý mạng truyền thống, thay vào đó họ chỉ cần truy cập vào giao diện quản trị của controller là có thể cấu hình cho toàn bộ hệ thống mạng.

- Khả năng mở rộng linh hoạt: Các chức năng của hệ thống mạng có thể được thêm vào thông qua việc lập trình và cài đặt các gói phần mềm lên controller. Nếu so với thiết bị phần cứng truyền thống trước đây thì các thiết bị này không thể mở rộng thêm chức năng.

- Các thiết lập liên quan đến việc truy cập có thể được thực hiện với từng thiết bị cụ thể trong mạng, điều này sẽ tăng khả năng bảo mật của toàn hệ thống, đồng thời giảm lưu lượng mạng không cần thiết.

Nhược điểm:

- Vai trò của controller là đặc biệt quan trọng nên nó cũng trở thành điểm dễ bị tấn công. Chính vì thế việc sử dụng cluster gồm nhiều controller khác nhau là cần thiết. Trong phần 3.1.2, nhóm tác giả đã tạo cluster controller và thử nghiệm khi 1 controller bị hỏng (nhóm đã thử ngắt 1 controller), lúc này hệ thống vẫn hoạt động được ổn định.

- Việc sử dụng nhiều controller sẽ tốn kém chi phí đầu tư và do dó không phù hợp với các doanh nghiệp hoặc đơn vị nhỏ. Trong khuôn khổ kinh phí cấp phát cho đề tài cấp Học viện, nhóm chưa thể triển khai thực tế tại khoa do vấn đề kinh phí triển khai thực tế quá lớn.

- Công nghệ còn khá mới và lượng khách hàng chưa cao nên việc tìm và sử dụng các thiết bị đắt tiền cũng như phần mềm còn khó khăn (do chưa nhiều đơn vị cung cấp giải pháp cho SDN). Vì vậy, nhóm nghiên cứu sử dụng việc mô phỏng các thiết bị và chạy thử nghiệm bằng các phần mềm Mininet và Onos.

4. KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã tổng quát hóa cấu trúc cũng như cơ chế hoạt động của hai hệ thống mạng truyền thống và mạng SDN. Dựa trên cấu trúc cũng như hoạt động của giao thức OpenFlow và hệ thống mạng SDN, nhóm đưa ra các giải pháp thiết kế vận hành cũng như khắc phục các lỗi, các lỗ hổng trong hệ thống mạng SDN, đưa ra được các lợi ích, các ưu nhược điểm của hệ thống mạng SDN khi triển khai. Việc tìm ra được các điểm yếu, điểm dễ bị tấn công trong mạng SDN, các mối nguy hiểm trong mạng và cách thức tấn công của các hacker giúp nhóm nghiên cứu đưa ra các giải pháp phòng chống trong mạng.

Nhóm nghiên cứu đã tiến hành thử nghiệm mô hình mạng SDN trên hệ điều hành Linux từ đó có kiến thức thực tế hơn, về quá trình xây dựng mạng, các tiến trình chạy thử nghiệm trên mô hình. Đây là một công nghệ mạng mới còn

đang trong giai đoạn phát triển và thử nghiệm nên chưa được áp dụng nhiều trong thực tế.

Nhóm đã nghiên cứu việc áp dụng mạng SDN vào thực tế tại Khoa Công nghệ thông tin, Học viện Nông nghiệp Việt Nam, tuy nhiên, việc đầu tư một hệ thống mới cần mua thêm nhiều thiết bị mới. Trong thời gian tới, nhóm sẽ tiếp tục tiến hành triển khai theo hướng áp dụng tại Khoa CNTT trong một dự án mới.

Chúng tôi sẽ cần nhiều nghiên cứu hơn nữa và triển khai trên môi trường mạng thực tế để đánh giá được chính xác các mối đe dọa an ninh trong mạng. Hầu hết các giải pháp bảo mật còn riêng lẻ, một số có khả thi nhưng cũng có nhiều biện pháp chưa đạt được kết quả như mong muốn. Cần có một nghiên cứu tổng thể để đưa ra một phương pháp toàn diện nhằm đạt kết quả tốt nhất trong bảo mật.

LỜI CẢM ƠN

Nhóm nghiên cứu xin chân thành cảm ơn Học viện Nông nghiệp Việt Nam đã tài trợ nghiên cứu này thông qua đề tài nghiên cứu khoa học cấp Học viện mã số T2020-10-49.

TÀI LIỆU THAM KHẢO

Baran P. (1964). On Distributed Communications Networks. IEEE Transactions on Communications Systems.

Ben Kepes (2015). The Software-defined Data Center in the Enterprise. Nimboxx

Dabbagh M., Hamdaoui B., Guizani M. & Rayes A. (2015). Software-Defiened Networking Security: Pros and Cons. IEEE Communications Magazine.

Feamster Rexford J. & Zegura E. (2014). The Road to SDN: An Intellectual History of Programmable Networks. SIGCOMM Comput. Commun. Rev.

ONF Solution Brief (2013). SDN Security considerations in the data center. Mike McBride, Editor.

Robert M.H. (2014). SDN and Security: why take over the hosts when you can take over the network. RSA Conference.

Sama M.R, Contreras L., Kaippallimalil J., Akiyoshi I., Haiyang Q. & Hui N. (2015). Software-defined control of the virtualized mobile packet core. IEEE Communications Magazine. 53(2): 107-115.

Thomas D.N. & Ken Gray (2013). SDN Software Defined Network. Chapter 13. O'Reilly.

White Paper (2013). Software-Defined Networking: Why we like it and how we are building on it. Cisco Inc.