BÀI THÍ NGHIỆM

7

MỘT SỐ ỨNG DỤNG MẠNG - FIREWALL

I. MỤC ĐÍCH THÍ NGHIỆM

Bài thí nghiệm này giúp sinh viên tạo một topology đơn giản sau đó sử dụng POX Controller để điều khiển mạng, sau đó thêm các flow-entry để tạo một firewall đơn giản.

II. THẢO LUẬN

Các bộ điều khiển mạng dùng trong mạng OpenFlow gồm có: bộ điều khiển mạng mặc định, POX, POX, SNAC (giao diện Web để quản lý các OpenFlow switch), Beacon (Java). Tuy nhiên bộ điều khiển mạng chính và đáng chú ý nhất là bộ điều khiển mạng POX. POX là bộ điều khiển mở, phát triển bằng ngôn ngữ Python, cấu trúc đơn giản, gọn nhẹ, dễ dàng tạo và thêm các module mới.

Mục đích chính của POX gồm:

- Cung cấp một platform cho phép người lập trình, phát triển mạng triển khai các ý tưởng mới trong lĩnh vực mạng, sử dụng phần cứng thật. Các nhà phát triển có thể điều khiển tất cả các kết nối trong mạng gồm có: forwarding, routing… Ngoài ra POX còn điều khiển cả flow-table trong switch.
- Cung cấp phần mềm quản lý mạng hữu ích cho các tổng đài (operator), gồm có việc quản lý tập trung cho tất cả các switch trong mạng, điều khiển truy nhập của người dùng.

Phương thức hoạt động của POX:

- POX chạy riêng rẽ trên một máy và quản lý việc chuyển tiếp các bản tin giữa nhiều switch khác nhau. Trong quá trình mô phỏng, giả lập POX được chạy trên cùng một máy với đồ hình mạng được tạo ra bởi Mininet.

- POX cung cấp các giao diện lập trình giúp cho nhà phát triển sử dụng dễ dàng lấy được thông tin về sự kiện trong mạng, can thiệp vào lưu lượng, điều khiển các quyết định chuyển mạch của switch và tạo được lưu lượng.
- Khi có flow mới xuất hiện trọng mạng, các gói đầu tiên sẽ được gửi đến bộ điều khiển mạng POX, tại đây có thể thực hiện được: quyết định xem khi nào sẽ chuyển tiếp các gói đi trong mạng, định tuyến cho gói tin, thu thập các thông tin thống kê, chỉnh sửa được gói trong flow đó hoặc có thể xem thêm được về các gói khác trong cùng flow để thu thập được thêm nhiều thông tin.

POX đơn thuần chỉ là một platform, việc điều khiển mạng được thực hiện bởi các phần tử chức năng trong POX gọi là POX component, mỗi component thực thi một chức năng riêng biệt như định tuyến, chuyển mạch, xác thực... Có thể chạy một lúc nhiều POX component với các chức năng điều khiển khác nhau làm cho việc điều khiển và quản lý mạng trở nên hoàn hảo hơn. Các ứng dụng trong bộ điều khiển mạng POX có thể kết hợp với nhận biết các sự kiện trong mạng (network event), can thiệp vào lưu lượng trong mạng, điều khiển định tuyến của các switch và tạo ra lưu lượng.

Các thành phần của POX:

- Các thành phần Stock: POX đi kèm với một số thành phần stock. Một số thành phần cung cấp chức năng cốt lõi, một số cung cấp tính năng tiện lợi, và một số chỉ là ví dụ. Sau đây là một số thành phần stock của POX.

✓ *forwarding.l2_learning*: Thành phần này làm cho OpenFlow switch hoạt động như một thiết bị switch layer 2. Thành phần này học địa chỉ lớp 2 (địa chỉ MAC), các flow được cài đặt được match với các nhiều trường của header càng tốt.

✓ *forwarding.l3_learning*: Thành phần này biến các OpenFlow switch thành thiết bị không hẳn là một router nhưng cũng không phải là một switch layer 2, nó là một switch layer 3. Nó được dùng làm ví dụ khá tốt về việc sử dụng thư viện gói tin của POX để kiểm tra và xây dựng các bản tin ARP request và ARP reply.

✓ *openflow.spanning_tree*: Được sử dụng để khám phá các phần tử để xây dựng đồ hình mạng, xây dựng một spanning tree và sau đó disable các port không nằm trên spanning tree. Kết quả là một topo không bị loop. Chú ý là thành phần này không liên quan tới Spanning Tree Protocol.

✓ *openflow.discovery*: được sử dụng để khám phá topo mạng bằng việc gửi các bản tin LLDP tới các switch.

✓ *web.webcore*: Thành phần webcore khởi động một webserver trong tiến trình POX. Các thành phần khác có thể giao tiếp với nó để cung cấp nội dung tĩnh hoặc động của riêng chúng.

✓ *proto.dhcpd*: Nó đơn giản là một DHCP server. Mặc nó lấy địa chỉ 192.168.0.254 và cấp IP cho các DHCP client trong dải địa chỉ 192.168.0.1 đến 192.168.0.253 với DNS server và Default Gateway chính là DHCP server.

- Phát triển các thành phần tự tạo: Là các thành phần mà ta tự phát triển cho POX. Trong một số trường hợp, chúng ta có thể phải xây dựng một thành phần làm được những gì chúng ta muốn.

III. YÊU CẦU VỀ THIẾT BỊ

Để thực hiện bài thí nghiêm này cần một PC chạy Ubuntu 12.04, đã cài đặt sẵn Mininet và POX Controller.

IV. TRÌNH TỰ THÍ NGHIÊM

Trong bài thí nghiệm, sinh viên sẽ tiến hành tạo topology đơn giản bằng Mininet sau đó chạy POX Controller để điều khiển mạng, cuối cùng thêm các flow-entry vào switch để tạo một firewall đơn giản.

❑ **1.** Tạo một topology đơn giản bằng Mininet

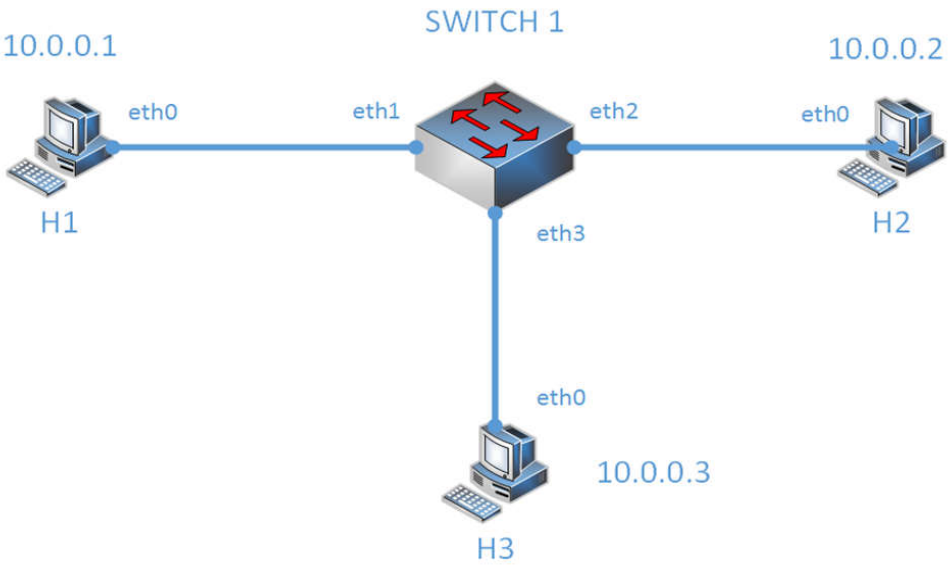
Chạy lệnh

```
sudo mn --topo single,3 --controller remote
```

Mininet sẽ tạo ra một topology đơn giản gồm 1 switch và 3 host.

```
user@machinename:~$ sudo mn --topo single,3 --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
```

```
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```



Hình 1. Topology 1 switch kết nối 3 host

Sau khi tạo topology thành công, sinh viên sẽ tiến hành chạy các module của POX để mạng có thể hoạt động ổn định bằng cách chạy các lệnh sau ở một cửa sổ terminal khác.

```
cd pox/
./pox.py openflow.discovery forwarding.l2_learning openflow.discovery
```

Lúc này, mạng có thể hoạt động ổn định, sinh viên tự kiểm tra bằng lệnh pingall.

❑ 2. Tạo một firewall đơn giản

Chức năng của tường lửa là ngăn không cho một gói tin từ các host nằm trong blacklist khi đi qua switch. Để làm được việc này, sinh viên cần sử dụng action DROP trong flow-entry. Khi một gói tin đến switch, nếu nó match với các trường trong flow-entry (địa chỉ nguồn/địa chỉ đích/loại giao thức), thì switch sẽ xử lý theo action trong flow-entry, ở đây là action DROP nên switch sẽ loại bỏ gói tin, không xử lý.

Trong thí nghiệm này, giả sử khi mạng đang hoạt động ổn định, khi phát hiện truy cập trái phép từ **host 1**, admintrastor sẽ lập tức install một flow-entry với trường địa chỉ

nguồn là địa chỉ của **host 1**, trường action là DROP, khi đó toàn bộ gói tin từ **host 1** đi qua **switch 1** sẽ bị DROP.

Để làm được việc này ta cần install một flow-entry vào **switch 1** với nội dung như sau

```
sudo ovs-ofctl add-flow s1 dl_type=0x0800, priority=65500, nw_src=10.0.0.1,
action=DROP
```

priority = 65500 là priority cao nhất, để tránh xung đột với các flow-entry khác trong switch.

Sau khi install flow-entry, **host 1** không thể ping đến bất kỳ host nào khác trong mạng.

V. KẾT LUẬN

Qua bài thí nghiệm này, sinh viên đã làm quen với POX, một controller rất phổ biến trong SDN, đồng thời cũng hiểu rõ cơ chế hoạt động của một Firewall đơn giản.

VI. CÂU HỎI KIỂM TRA

1. Khi topology có loop, trên POX cần chạy module gì để mạng có thể hoạt động ổn định ?

……
……
……
……