

Firewall Design with SDN

The following is the method of installing the software to define the network platform Mininet.

1. Use following commands to install Mininet in your Linux machine:

```
# git clone git://github.com/mininet/mininet
# cd mininet/util/
# ./install.sh -a
```

2. After a few minutes wait, following tools will be set:

mininet of dissector oflops oftest openflow pox

3. Successfully installation information:

```
libtool: install: /usr/bin/install -c cbench /usr/local/bi
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/mininet/util/oflops/cbench'
make[1]: Leaving directory `/mininet/util/oflops/cbench'
Making install in doc
make[1]: Entering directory `/mininet/util/oflops/doc'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/mininet/util/oflops/doc'
Enjoy Mininet!
```

4. Then you can run mn for test:

```
root@kali:~/mininet/util# mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

No connection

After that, we through the program topoCreate.py to create s1, s2, s3, s4 four hosts, as well as switches s1, note that this script is in the line of code

```
net = Mininet (topo=topo, controller= lambda  
name:RemoteController(name,ip='0.0.0.0'),link=TCLink)
```

This inside the remoteController ip I was 0.0.0.0, according to the need to modify their own.

Run our program to get the topology of the line:

```
#python topoCreate.py
```

```
root@tian01:/home/tian/michael/mininet# python topoCreate.py
Creating network and run simple performance test
*** Creating network
*** Adding controller
Unable to contact the remote controller at 0.0.0.0:6653
Unable to contact the remote controller at 0.0.0.0:6653
Setting remote controller to 0.0.0.0:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s0
*** Adding links:
(20.00Mbit 5ms delay) (20.00Mbit 5ms delay) (h1, s0) (20.00Mbit 5ms delay) (20.00Mbit 5ms delay) (h2, s0) (20.00Mbit 5ms delay) (20.00Mbit 5ms delay) (h3, s0) (20.00Mbit 5ms delay)
(20.00Mbit 5ms delay) (h4, s0)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s0 ... (20.00Mbit 5ms delay) (20.00Mbit 5ms delay) (20.00Mbit 5ms delay)
Dumping host connections
h1 h1-eth0:s0-eth1
h2 h2-eth0:s0-eth2
h3 h3-eth0:s0-eth3
h4 h4-eth0:s0-eth4
*** Starting CLI:
mininet> |
```

Result:

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=24846>
<Host h2: h2-eth0:10.0.0.2 pid=24848>
<Host h3: h3-eth0:10.0.0.3 pid=24850>
<Host h4: h4-eth0:10.0.0.4 pid=24852>
<OVSSwitch s0: lo:127.0.0.1,s0-eth1:None,s0-eth2:None,s0-eth3:None,s0-eth4:None pid=24857>
<RemoteController c0: 0.0.0.0:6653 pid=24838>
```

Then we re-open a use POX to create a controller, the specific method is:

In the mininet below there is a pox folder, run:

```
#python pox.py openflow.of_01 -port = 6653 pox.forwarding.l2_pairs.
```

For example,

```
root@tian01:/mininet/util/pox# python pox.py openflow.of_01 -port=6653 pox.forwarding.l2_pairs
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:forwarding.l2_pairs:Pair-Learning switch running.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[e2-13-d6-7d-a6-4d 1] connected
```

You can find the switch has been connected. Test connectivity (both pingable)

```

mininet> h1 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=20.3 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=20.1 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 20.107/20.246/20.385/0.130 ms
mininet> h1 ping -c2 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=85.2 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=20.1 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 20.115/52.701/85.287/32.586 ms
mininet> h1 ping -c2 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=109 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=20.1 ms

--- 10.0.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 20.116/64.563/109.010/44.447 ms
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=24846>
<Host h2: h2-eth0:10.0.0.2 pid=24848>
<Host h3: h3-eth0:10.0.0.3 pid=24850>
<Host h4: h4-eth0:10.0.0.4 pid=24852>
<OVSSwitch s0: lo:127.0.0.1,s0-eth1:None,s0-eth2:None,s0-eth3:None,s0-eth4:None pid=24857>
<RemoteController c0: 0.0.0.0:6653 pid=24838>
mininet> /Users/shenzhenyu
*** Unknown command: /Users/shenzhenyu
mininet>
mininet>
mininet>
mininet> dpctl dump-flows
*** s0
-----
OVS_FLOW reply (xid=0x4):
cookie=0x0, duration=170.708s, table=0, n_packets=3, n_bytes=238, idle_age=165, dl_src=00:00:00:00:01,dl_dst=00:00:00:00:04 actions=output:4
cookie=0x0, duration=278.306s, table=0, n_packets=13, n_bytes=1162, idle_age=178, dl_src=00:00:00:00:01,dl_dst=00:00:00:00:02 actions=output:2
cookie=0x0, duration=175.667s, table=0, n_packets=3, n_bytes=238, idle_age=174, dl_src=00:00:00:00:03,dl_dst=00:00:00:00:01 actions=output:1
cookie=0x0, duration=278.267s, table=0, n_packets=14, n_bytes=1204, idle_age=178, dl_src=00:00:00:00:02,dl_dst=00:00:00:00:01 actions=output:1
cookie=0x0, duration=175.705s, table=0, n_packets=2, n_bytes=196, idle_age=174, dl_src=00:00:00:00:01,dl_dst=00:00:00:00:03 actions=output:3
cookie=0x0, duration=170.071s, table=0, n_packets=4, n_bytes=200, idle_age=102, dl_src=00:00:00:00:04,dl_dst=00:00:00:00:02 actions=output:1
mininet>

```

Run the firewall program:

You need two terminals to run misc_pfirewall.py and misc respectively

Terminal 1:

```
root@tian01:/mininet/util/pox# python pox.py openflow.of_01 -port=6653 pox.forwarding.l2_pairs misc_pfirewall.py
```

```

INFO:misc_pfirewall:Traffic to h1 is 532 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.09576 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 24982 bytes
INFO:misc_pfirewall:Traffic to h1 is 4.49676 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 181764 bytes
INFO:misc_pfirewall:Traffic to h1 is 32.71752 bandwidth
INFO:misc_pfirewall:Overload: Packets from h4 are dropped
INFO:misc_pfirewall:sending packets successfully
INFO:misc_pfirewall:Traffic to h1 is 78 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.01404 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 0 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.0 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 78 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.01404 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 7738 bytes
INFO:misc_pfirewall:Traffic to h1 is 1.39284 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 181764 bytes
INFO:misc_pfirewall:Traffic to h1 is 32.71752 bandwidth
INFO:misc_pfirewall:Overload: Packets from h4 are dropped
INFO:misc_pfirewall:Traffic to h1 is 21798 bytes
INFO:misc_pfirewall:Traffic to h1 is 3.92364 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 192 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.03456 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 0 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.0 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 0 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.0 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 0 bytes
INFO:misc_pfirewall:Traffic to h1 is 0.0 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 84034 bytes
INFO:misc_pfirewall:Traffic to h1 is 15.12612 bandwidth
INFO:misc_pfirewall:Traffic to h1 is 146316 bytes
INFO:misc_pfirewall:Traffic to h1 is 26.33688 bandwidth
INFO:misc_pfirewall:Overload: Packets from h2 are dropped
INFO:misc_pfirewall:sending packets successfully

```

Terminal 2:

Used to send iperf command

```

mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['19.1 Mbits/sec', '23.2 Mbits/sec']
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['19.1 Mbits/sec', '23.2 Mbits/sec']
mininet>

```