Exercise: Pyretic Debugging

HINT: You might have to use the "$ dpctl dump-flows tcp:127.0.0.1:6634" or "mininet> dpctl dump-flows" command frequently.

-
- In this debugging exercise, we take solutions available in the Internet for the gardenwall problem and try to fix bugs in it.
- The basic solution is taken from the Internet [1], test if it is able to block h1 when "infected". Note that we will only use the "infected == True" for this exercise.
  - Copy the above code into /home/mininet/pyretic/pyretic/examples as pyretic_gardenwall_internetsolution.py
  - start controller (in /home/mininet/pyretic folder):

```
$pyretic.py pyretic.examples.pyretic_gardenwall_internetsolution
```

```
mininet@mininet-vm:~/pyretic$ pyretic.py pyretic.examples.pyretic_gardenwall_int
ernetsolution
Couldn't import pydot: dot visualization will not be possible.
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
Connected to pyretic frontend.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
```

  - start mininet:

```
$ sudo mn --controller=remote --topo=single,3 --mac --arp
```

```
mininet@mininet-vm:~$ sudo mn --mac --arp --switch ovsk --link=tc --topo=single,
3 --controller=remote,ip=127.0.0.1
*** 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:
```

  - check h1 ping h2

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

- Now infect h1 (in /home/mininet/pyretic/pyretic/kinetic folder):

```
    $ python json_sender.py -n infected -l True
--flow="{srcmac=00:00:00:00:00:01}" -a 127.0.0.1 -p 50001
```

```
mininet@mininet-vm:~/pyretic/pyretic/kinetic$ python json_sender.py -n infected
-l True --flow="{srcmac=00:00:00:00:00:01}" -a 127.0.0.1 -p 50001

Flow_Str = {srcmac=00:00:00:00:00:01}

Data Payload = {'dstip': None, 'protocol': None, 'srcmac': '00:00:00:00:00:01',
'tos': None, 'vlan_pcp': None, 'dstmac': None, 'inport': None, 'switch': None, '
ethtype': None, 'srcip': None, 'dstport': None, 'srcport': None, 'vlan_id': None
}

return: None
```

check h1 ping h2. We should be able to observe that this traffic is blocked.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
```

Now, we move on to the debugging part

---

check h2 ping h3, what happens?

It is also be blocked

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
```

Now, modify the given code to allow h2 traffic to pass through to h3, when h1 is "infected".

The result is in the following, and the code is in the attachment.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X h3
h3 -> X h2
*** Results: 66% dropped (2/6 received)
mininet>
```

Controller screen shot:

```
mininet@mininet-vm:~/pyretic$ pyretic.py pyretic.examples.pyretic_gardenwall_in
ernetsolution
Couldn't import pydot: dot visualization will not be possible.
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
Connected to pyretic frontend.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
Event arrived.
   Flow:  {'srcmac': 00:00:00:00:00:01}
   Event name:  infected
   Value:  True
{{'srcmac': 00:00:00:00:00:01}: {'infected': 'True'}}
Infected
union:
    sequential:
        if
            match: ('srcmac', 00:00:00:00:00:01)
        then
            drop
        else
            identity
        if
            match: ('srcmac', 00:00:00:00:00:03)
        then
            identity
        else
            match: ('dstip', 10.0.0.3)
```

Relevant event screen shot:

```
mininet@mininet-vm:~/pyretic/pyretic/kinetic$ python json_sender.py -n infected
-l True --flow="{srcmac=00:00:00:00:00:01}" -a 127.0.0.1 -p 50001

Flow_Str = {srcmac=00:00:00:00:00:01}

Data Payload = {'dstip': None, 'protocol': None, 'srcmac': '00:00:00:00:00:01',
'tos': None, 'vlan_pcp': None, 'dstmac': None, 'inport': None, 'switch': None, '
ethtype': None, 'srcip': None, 'dstport': None, 'srcport': None, 'vlan_id': None
}

return: None
```

- Now, check if the "exempt" case is working fine too (i.e if h1 is "infected" and "exempt", send it to h3 instead of blocking it)

```
  $ python json_sender.py -n exempt -l True
--flow="{srcmac=00:00:00:00:00:01}" -a 127.0.0.1 -p 50001
```

Event screen shot:

```
mininet@mininet-vm:~/pyretic/pyretic/kinetic$ python json_sender.py -n exempt -l
 True --flow="{srcmac=00:00:00:00:00:01}" -a 127.0.0.1 -p 50001

Flow_Str = {srcmac=00:00:00:00:00:01}

Data Payload = {'dstip': None, 'protocol': None, 'srcmac': '00:00:00:00:00:01',
'tos': None, 'vlan_pcp': None, 'dstmac': None, 'inport': None, 'switch': None, '
ethtype': None, 'srcip': None, 'dstport': None, 'srcport': None, 'vlan_id': None
}

return: None
```

Controller screen shot:

```
Event arrived.
   Flow:  {'srcmac': 00:00:00:00:00:01}
   Event name:  exempt
   Value:  True
{{'srcmac': 00:00:00:00:00:01}: {'infected': 'True', 'exempt': 'True'}}
Install garden wall
union:
    sequential:
        match: ('dstmac', 00:00:00:00:00:02)
        modify: ('dstmac', 00:00:00:00:00:03)
        identity
```

Mininet screen shot:

```
*** Results: 100% dropped (0/6 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> X h3
h2 -> X X
h3 -> h1 X
*** Results: 66% dropped (2/6 received)
```

The code is in the attachment.