Hopefully part 2 wasn’t too bad...
Part 3

- You can hardcoded who-is-where in `cores21_setup`.
- Run `links` in the Mininet console to see who’s where.
- If your `pingall` fails, make sure that you flood ARP. (Why?)

```
[h1@10.0.1.10/24]--{s1}--\  
[h2@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]  
[h3@10.0.3.30/24]--{s3}--/  
   |                     
   |                     
[hnotrust1@172.16.10.100/24]
```
Part 4

- Must not hardcode who-is-where.
- Learn by backward learning --- learn who-is-where when we hear from them.

```
[h10@10.0.1.10/24]--{s1}--
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/
    |   
[hnnotrust1@172.16.10.100/24]
```
Part 4 - h10 ping h20

I want to ping 10.0.2.20. That’s not in my subnet, so I know that should go through 10.0.1.1.

[h10@10.0.1.10/24]--{s1}--\n[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/    |
                           [hnotrust1@172.16.10.100/24]
Part 4 - h10 ping h20

ARP REQUEST:
Who is 10.0.1.1?
Tell 10.0.1.10.

[h10@10.0.1.10/24]--{s1}--
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/
|    |
[hnnotrust1@172.16.10.100/24]
I just got an ARP request from 10.0.1.0/24 through port 1. So in the future, I will remember to forward traffic to 10.0.1.0/24 through port 1.

[Installs a ofp_flow_mod rule]
I'm going to handle traffic for s1.
Part 4 - h10 ping h20

10.0.1.1 is at de:ad:be:ef:ca:fe (I just made that up, but I replied so that’s me 🙋).
Ok, I got the ARP reply. I think cores21 has 10.0.1.1. In the future, I will send out-of-network traffic through cores21.

[h10@10.0.1.10/24]--{s1}--\ 
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24] 
[h30@10.0.3.30/24]--{s3}--/    | 
| 
| [hnotrust1@172.16.10.100/24]
Part 4 - h10 ping h20

Ping 10.0.2.20

[h10@10.0.1.10/24]--{s1}--\n[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/    |
    |
    [hnotrust1@172.16.10.100/24]
I just got ICMP traffic to 10.0.2.20, but I don’t know where it’s at. I’ll just drop it.
Part 4 - h10 ping h20

[Part 4 - h10 ping h20]---{s1}---
[h20@10.0.2.20/24]---{s2}---{cores21}---{dcs31}---[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]---{s3}---/

[hnnotrust1@172.16.10.100/24]

😠

okay.
[times out]
I want to ping 10.0.1.10. That’s not in my subnet, so I know that should go through 10.0.2.1.
Part 4 - h20 ping h10

ARP REQUEST:
Who is 10.0.2.1?
Tell 10.0.2.20
I just got an ARP request from 10.0.2.20/24 through port 2. So in the future, I will remember to forward traffic to 10.0.2.20/24 through port 2.

[Installs a `ofp_flow_mod` rule]
Part 4 - h20 ping h10

ARP REPLY:
10.0.2.1 is me,
de:ad:be:ef:ca:fe.

[h10@10.0.1.10/24]--{s1}--\  
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]  
[h30@10.0.3.30/24]--{s3}--/  
  [hnotrust1@172.16.10.100/24]
Ok, I got the ARP reply. I think cores21 has 10.0.2.1. In the future, I will send out-of-network traffic through cores21.
Part 4 - h20 ping h10

[10@0.1.10/24]--{s1}--\n[10@0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@0.4.10/24]
[10@0.3.30/24]--{s3}--/
                             |
                             v
[notrust1@172.16.10.100/24]

Ping 10.0.1.10
I just got ICMP traffic to 10.0.1.10. My rules tell me to forward it thru port 1.
I got the ICMP request. I'll respond.
Part 4 - h20 ping h10

I just got ICMP traffic to 10.0.2.20. My rules tell me to forward it thru port 2.
Part 4 - h20 ping h10

[host1@10.0.1.10/24]--{s1}--\
[host2@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[host3@10.0.3.30/24]--{s3}--/

|

trust1@172.16.10.100/24

😊 noice
Part 4 Summary

- You need to block all IP traffic between server 1 and untrusted host
- You need to block all ICMP traffic from untrusted host
- You need to change _handle_PacketIn function
  - You need to handle ARP packets and add it to the routing table
  - You need to send the arp ack back to the sender. You can import arp() from pox
  - While sending the arp reply back, you can hardcode the mac address of cores21. You don’t actually need to find it because it doesn’t matter.
Q&A, Extra OH