

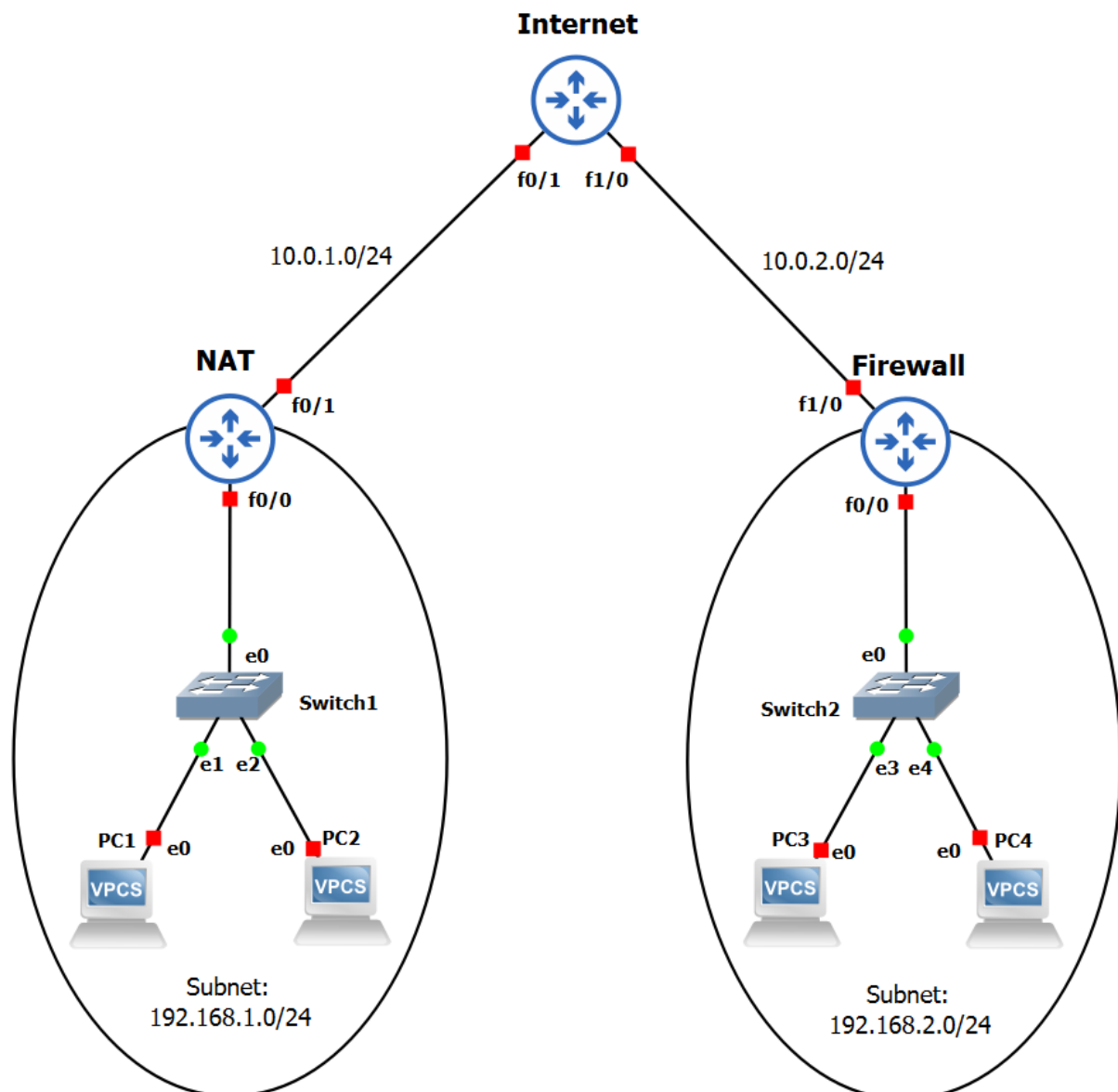
# ELEC-H417: Lab 5 - Middleboxes

## Objectives

**Middleboxes** define the family of devices that transform, filter and manipulate network traffic for any purpose other than packet forwarding. Firewalls and NATs are common middleboxes.

- Configure and understand simple **firewall** rules
- Configure and understand a small **NAT (Network Address Translation)**

## Topology



The topology is already configured with the following values:

- **Hosts**
  - PC1 (192.168.1.2/24 gateway 192.168.1.1)
  - PC2 (192.168.1.3/24 gateway 192.168.1.1)
  - PC3 (192.168.2.2/24 gateway 192.168.2.1)
  - PC3 (192.168.2.3/24 gateway 192.168.2.1)
- **Routers**
  - NAT
    - f0/0 192.168.1.1
    - f0/1 10.0.1.1
  - Firewall
    - f0/0 192.168.2.1
    - f1/0 10.0.2.2
  - Internet
    - f0/1 10.0.1.3
    - f1/0 10.0.2.3

## Mission 0: Topology verification (not needed in the report)

---

Quickly verify that everything is loaded correctly by using pings (i.e. ping PC2, PC3 and PC4 from PC1).

To clarify, “Firewall”, “NAT” and “Internet” are 3 CISCO routers (the same that you have used during previous labs) and do nothing special for the moment. During Mission 1, you will have to configure a firewall on the router called “Firewall”. During Mission 2, you will have to configure a NAT on the router called “NAT”. And finally, the last router called “Internet” is just a normal router that represents the internet network.

# Mission 1: NAT

---

## Mission 1.1: Configuration



Before doing the mission, from your actual knowledge:

- *What is a NAT ?*
- *How does it work ?*

*NB: At this step, you can be wrong it won't be evaluated*

First you need to indicate the two interfaces used by the NAT as well as the direction (i.e. inside and outside). In our topology, interface f0/0 is linked to the inside part of the NAT (inside subnet).

```
NAT(config)# interface FastEthernet0/0
NAT(config-if)# ip nat inside
```

And interface f0/1 is linked to the outside part of the NAT (outside world, aka. internet)

```
NAT(config)# interface FastEthernet0/1
NAT(config-if)# ip nat outside
```

Then you can add some translation entries in the NAT table as follows:

```
NAT(config)# ip nat inside source static udp 192.168.1.2 80 10.0.1.1 1111
NAT(config)# ip nat inside source static udp 192.168.1.3 80 10.0.1.1 2222
```

You can check your NAT translation table with the following command:

```
NAT# show ip nat translations
```

Another interesting commands is the following on:

```
NAT# show ip nat statistics
```

For more information, do not hesitate to check the [documentation](#).

## Mission 1.2: Analysis

Now that your NAT is configured with 2 simple entries (or maybe more for the most curious students 😊). You can try the following pings and observe what happens on **Wireshark** captures on the link before AND after the NAT (i.e. capture on both interfaces of the NAT):

- Ping from PC1 port 8080 to PC4: `ping 192.168.2.3 -P 17 -p 80 -s 8080`
- Ping from PC1 port 80 to PC4: `ping 192.168.2.3 -P 17 -p 80 -s 80`
- Ping from PC2 port 80 to PC4: `ping 192.168.2.3 -P 17 -p 80 -s 80`
- Ping from PC4 to PC1 on port 80 : `ping 192.168.1.2 -P 17 -p 80 -s 80`
- Ping from PC4 to PC1 on port 1111: `ping 192.168.1.2 -P 17 -p 1111 -s 80`
- Ping from PC4 to NAT on port 1111: `ping 10.0.1.1 -P 17 -p 1111 -s 80`
- Ping from PC4 to NAT on port 2222: `ping 10.0.1.1 -P 17 -p 2222 -s 80`

The different parameters for the ping commands mean (see documentation: `ping ?`)

- `-P` : specify the protocol (e.g. 17=UDP)
- `-p` : specify the destination port
- `-s` : specify the source port



What can you observe ? Explain the results obtained.  
With that in mind explain how



After doing the mission, what have you learned ?

- What is a NAT ?*
- How does it work ?*

NB: You can answer “nothing” if your previous answer was perfect 😊



What is the use of a NAT ? What are the **advantages** and **disadvantages** ?

## Mission 2: Firewall



Before doing the mission, from your actual knowledge:

- *What is a Firewall ?*
- *How does it work ?*

*NB: At this step, you can be wrong it won't be evaluated*

During this mission, you will learn to create network traffic rules on a router to act as a firewall.

### Mission 2.1: Understanding the commands

Here is a small summary of how to configure rules for the firewall, but do not hesitate to check the [documentation](#).

#### Creating an ACL

First of all, you must create an ACL (Access control list). This is a list of rules that determine whether the packet should be accepted (=permit) or rejected (=deny) based on different parameters such as source IP, destination IP, protocol, etc.

```
Firewall(config)# ip access-list extended blacklist
Firewall(config-ext-nacl)# deny icmp host 192.168.1.2 host 192.168.2.3
Firewall(config-ext-nacl)# permit ip any any
```

You can verify the different ACL lists created and their rules as follows:

```
Firewall# show access-lists
Extended IP access list blacklist
 10 deny icmp host 192.168.1.2 host 192.168.2.3
 20 permit ip any any
```

This simple example of ACL, will reject every ping from PC1 to PC4 and accept all the other communication.

#### Rules

To explain a bit the commands; you first create an ACL called "blacklist" (you can choose the name that you want). Afterwards, each line is a rule that you add to this ACL. These

rules can be generalized as:

```
Firewall(config-ext-nacl)# <order> [permit/deny] [protocol] [source] [destination] <eq [port]>
```

As a reminder, a firewall is an ordered list of rules to follow when a packet arrived. Therefore, it will check the first rule, if does nothing, it check the second rule and so on... The parameter `<order>` is an **optionnal** integer indicating where to place this rule in the list. If nothing is specified, it will add the new rule at the end of the list.

The third parameter is the `[protocol]`. For example, it can be ICMP (the default protocol for ping), UDP, TCP, or more generally IP.

Then the `[source]` and `[destination]` parameter can either be:

- `host [IP]` : A specific host  
e.g. `host 192.168.1.2`
- `[IP] [inverse mask]` : A subnet (be careful, it is a inverse mask... “1” means to not check this bit).  
e.g. `192.168.1.0 0.0.0.255`
- `any` : Which will match with anything

You can append your command with an **optionnal** parameter `eq [port]` which allows to filter for a specific destination port. For example, the following rule deny UDP communication from PC1 to the port 80 of PC4:

```
Firewall(config-ext-nacl)# deny udp host 192.168.1.2 host 192.168.2.3 eq 80
```

### Adding or removing rules

Afterwards, you can add new rules to an existing ACL as:

```
Firewall# configure terminal
Firewall(config)# ip access-list extended blacklist
Firewall(config-ext-nacl)# 15 deny ip 192.168.1.0 0.0.0.255 any
Firewall(config-ext-nacl)# end
Firewall# show access-list
Extended IP access list blacklist
  10 deny icmp host 192.168.1.2 host 192.168.2.3
  15 deny ip 192.168.1.0 0.0.0.255 any
  20 permit ip any any
```

And you can remove a rule from an existing ACL with the keyword “**no**” followed by the whole rule or simply its ID (=order number):

```
Firewall# configure terminal
Firewall(config)# ip access-list extended blacklist
Firewall(config-ext-nacl)# no 15
Firewall(config-ext-nacl)# end
Firewall# show access-list
Extended IP access list blacklist
    10 deny icmp host 192.168.1.2 host 192.168.2.3
    20 permit ip any any
```

## Enable or Disable an ACL

Now that you have your ACL, you still need to enable it on an interface of the router.


```
Firewall(config)# interface FastEthernet1/0
Firewall(config-if)# ip access-group blacklist in
```

Similarly, you can disable it with the keyword **“no”**.

```
Firewall(config)# interface FastEthernet1/0
Firewall(config-if)# no ip access-group blacklist in
```

To keyword **“in”** means that the ACL's rules are applied on the traffic arriving to the router through this interface. You can put the keyword **“out”** to applied the rules on the traffic leaving the router by this interface.

## Mission 2.2: Create your firewall

 *Is it better to create a firewall which by default blocks everything and accepts according to some rules (whitelist) or a firewall which accepts everything but refuses according to some rules (blacklist) ? Make a brief pros and cons of those two possibilities.*

Your mission will be to create and apply an efficient ACL that fulfills these conditions:

- Accept ping (ICMP) from PC1 but not PC2
- Refuse UDP communication unless intended for port 80 of PC4
- PC3 accepts all UDP communication from subnet with PC1 and PC2



Verify that your ACL is working as expected by using suitable ping commands and provide the output of `show access-list` command.



After doing the mission, what have you learned ?

- *What is a Firewall ?*
- *How does it work ?*

NB: You can answer “nothing” if your previous answer was perfect 😊

## Mission 2.3: Protect yourself against data exfiltration

As a last mission, let's create and apply a new ACL (still on the router's “Firewall”) preventing any possible leak of information from the 192.168.2.0 subnet (with PC3 and PC4). In other words, block all outgoing communication from the 192.168.2.0 subnet.



Verify that your ACL is working as expected by using suitable ping commands and provide the output of `show access-list` command.

Have you try to ping PC4 from PC1 ? What happend ?



*Do you have any idea why it might be interesting to block outgoing connections ?*

## Final Mission

Carefully reread the course and labs (as well as additional sources) to verify that you have understood the main concepts and mechanisms of network communication.

`Good luck for your exams`