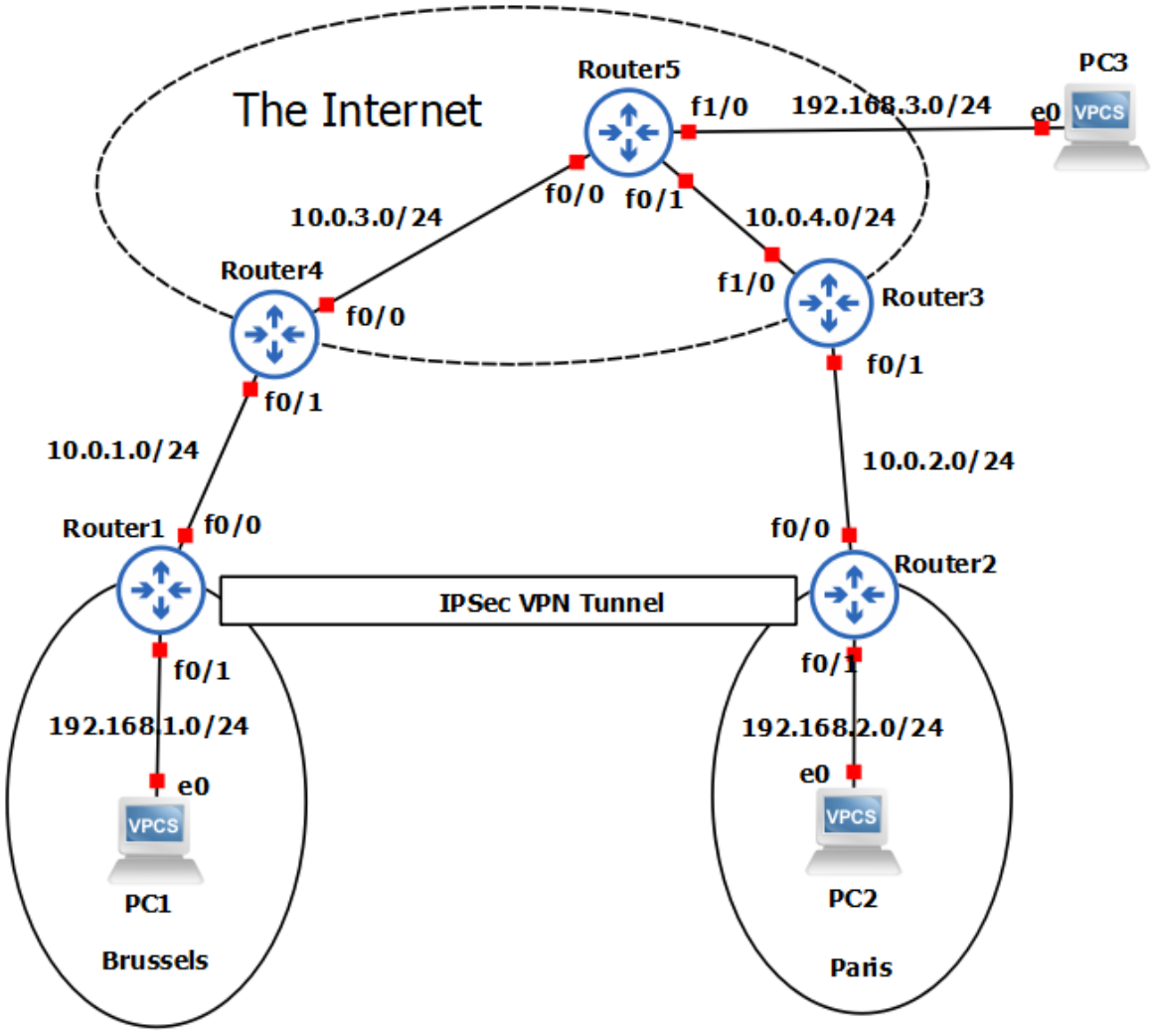
ELEC-H417: Lab 4 - VPN

Objectives

- Configure L3 (IP-in-IP) VPN between two sites
- Understand IPSec options and phase 1 and phase 2 negociations

Topology



The topology is already configured with the following values:

- **Guest hosts**
  - PC1 (192.168.1.2/24 gateway 192.168.2.1), aka Brussels

    - PC2 (192.168.2.2/24 gateway 192.168.2.1), aka Paris
    - PC3 (192.168.3.2/24 gateway 192.168.3.1)
- **Routers**
    - Router1
        - f0/1 192.168.1.1
        - f0/0 10.0.1.1
    - Router2
        - f0/1 192.168.2.1
        - f0/0 10.0.2.1
    - Router3, Router4, Router5 (the Internet)
    - Various subnets
    - RIP dynamic routing for routes propagation

Mission 1: Traffic observation

Using Wireshark:

1. Activate traffic logging between Router1 and Router4. Ping from PC1 to PC3 and validate that the traffic is in clear.
2. Do likewise for a ping between PC1 and PC2. Validate that the traffic is in clear.

Mission 2: Site-to-Site IPSec VPN

We will consider here site-to-site L3 VPN (IPSec) with pre-shared passwords in IPSec phase 1.
The IKE policy selected (e.g., from a prior crypto analysis or recommandation) is the following:

```
encryption algorithm: AES-256
hash algorithm: SHA1
authentication method: Pre-Shared Key
Diffie-Hellman group: #5
lifetime: 86400 seconds
```

In practice, several other options are available, depending on the hardware/software you will
have at hand (CISCO, Juniper, Palo Alto, Huawei,...).

In practice, a VPN induces a loss of throughput of ~25% due to overheads, processing, etc.
This is why some implementation are hardware accelerated by means of cryptoprocessors in
the equipment (e.g. AES is often found). These side processors are also available in commodity equipments (e.g. Apple computers, INTEL processors, ...).

IPSec Phase 1 (aka IKE)

IKE (*Internet Key Exchange*) exists only to establish a SAs (Security Association) between two IPsec end points. Remember that IPSec allows to have specific parameters for each pair of source and
destination.

As a first step, IKE negotiates a SA (an ISAKMP SA) relationship with the peer by **authenticating**. This is reflected by means of a ISAKMP (*Internet Security Association and Key Management Protocol*) Phase 1 *policy*:

```
R1(config)# crypto isakmp policy 1
R1(config-isakmp)# encryption aes
R1(config-isakmp)# hash sha
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 2
R1(config-isakmp)# lifetime 86400
R1(config-isakmp)# exit
```

Note that the policy is not (yet) related to a specific peer. It describes what parameters are in use for the encryption, HMAC, key exchange, etc... Several policies can be described and they will be tried one by one to initiate the VPN tunnel (until one actually works with the remote peer).

In a second step, let us now set the **pre-shared password** ("*unicorn*") for our destination:

```
R1(config)# crypto isakmp key unicorn address 10.0.2.1
```

All parameters for Phase 1 are now set. What happens in phase 1 ? Well, when detecting a packet subject to crypto transformation (see below), the destination is

looked up. Then, the password (or certificate) for that destination is retrieved from the local database. Next, each policy is used (one after one) to try to establish a Phase 1 with the remote peer. When one is IPSec Phase 1 (aka IKE) accepted a SA (IPSec security assiciation) is created.

Now let's move to phase 2 to configure how packets will be transformed.

ISAKMP Phase 2 (aka crypto transformation and access control)

To configure IPSec we need to setup the following in order:

- Create extended ACL (*access control list*) to determine what packets are subject (ornot) to transformation. This allows to separate between the normal traffic and the VPN traffic.
- Create IPSec Transform to configure how packets will be ciphered, MAC'ed, etc..
- Create Crypto Map to link a specific destination with a specific transform and filtered with the ACL (it is a glue of the previous steps)
- Apply crypto map to the public interface to tell the router to analyze every packet passing by that interface for a possible crypto transformation.

Now let us implement each of the above steps in the CISCO routers:

- **Create extended ACL**:
  First step is to create an access-list and define the traffic we would like the router to pass through the VPN tunnel. In this example, it would be traffic from one network to the other, 10.10.10.0/24 to 20.20.20.0/24. Access-lists that define VPN traffic are sometimes called **crypto access-list** or **interesting traffic access-list** (note the wildcard notation 0.0.0.255 which is the bit-invert notation of netmask for the matching).

  ```
  R1(config)# ip access-list extended VPN-TRAFFIC
  R1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
  ```

  The traffic from 192.168.1.0/24 to 192.168.2.0/24 will be subject to tagging and processing in
  the pipeline called "VPN-TRAFFIC" (you can put here any name you like).

- **Create IPSec Transform**:

  There are defined the mode (**AH** or **ESP**) and the ciphering parameters.

```
R1(config)# crypto ipsec transform-set MYTS esp-aes esp-sha-hmac
```

A transformation set is prepared and can be applied to the packets. Nothing special here.

- **Create Crypto Map**:

  The Crypto map is the last step of our setup and connects the previously defined ISAKMP and IPSec configuration together:

```
R1(config)# crypto map CMAP 10 ipsec-isakmp
R1(config-crypto-map)# set peer 10.0.2.1
R1(config-crypto-map)# set transform-set MYTS
R1(config-crypto-map)# match address VPN-TRAFFIC
```

  The *crypto map* bundles everything. It tells to follow these steps for each packet:

  1. If a packets is tagged as VPN-TRAFFIC
  2. Apply the transformation MYTS (ciphering, MAC'ing)
  3. Send it to the tunnel endpoint 10.0.2.1. Note that this last step will trigger the Phase 1 to
     authenticate and negotiate a session key between the two tunnel peers (everything makes
     sense now).

- **Apply Crypto Map**:

  Finally, we must apply the crypto map to a (or several) network interface.

```
Router1(config)# interface fastEthernet 0/0
Router1(config-if)# crypto map CMAP
```

  This simply means that any packet flowing in this interface will be inspected as per the rules defined in the crypto map CMAP. One last word: this must obviously be configured on both ends of the tunnel.

Mission 3: Validate

Traffic on the internet links

Capture the traffic with Wireshark and ping the remote hosts. First from PC1 to PC3 (no VPN tunnel) on the line between Router1 and Router4. You should observe ICMP ping message sent in clear.

Next ping two tunnel ends (PC1 and PC2). This time, the packets are subject to the transform rule and will and will be encapsulated in IPSec ESP. Note that the first ping packets may be lost (why ?). The reason is two-fold:

1. ARP requests (as usual)
2. Time needed to negotiate the phase 1 and create the corresponding IPSec SA.

Validation of the tunnel (IKE Policies, transformations)

It is possible to display the current configuration via the following commands:

```
R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 1
 encryption algorithm: AES - Advanced Encryption Standard (128 bit keys).
 hash algorithm: Secure Hash Standard
 authentication method: Pre-Shared Key
 Diffie-Hellman group: #5 (1536 bit)
 lifetime: 86400 seconds, no volume limit
```

and the active crypto transformations:

```
Router1# show crypto isakmp sa
dst src state conn-id slot status
10.0.2.1 10.0.1.1 QM_IDLE 1 0 ACTIVE

Router1# show crypto isakmp peers
Peer: 10.0.2.1 Port: 500 Local: 10.0.1.1
 Phase1 id: 10.0.2.1
 Router1#show crypto ipsec sa
 interface: FastEthernet0/0
 Crypto map tag: CMAP, local addr 10.0.1.1
 protected vrf: (none)
 local ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
 remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
 current_peer 10.0.2.1 port 500
 PERMIT, flags={origin_is_acl,}
 #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
 #pkts decaps: 3, #pkts decrypt: 3, #pkts verify: 3
 #pkts compressed: 0, #pkts decompressed: 0
 #pkts not compressed: 0, #pkts compr. failed: 0
 #pkts not decompressed: 0, #pkts decompress failed: 0
 #send errors 1, #recv errors 0
 local crypto endpt.: 10.0.1.1, remote crypto endpt.: 10.0.2.1
```

```
 path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
 current outbound spi: 0x4F33CBD1(1328794577)
 inbound esp sas:
 spi: 0x1FA02880(530589824)
 transform: esp-aes esp-sha-hmac ,
 in use settings ={Tunnel, }
 conn id: 2001, flow_id: SW:1, crypto map: CMAP
 sa timing: remaining key lifetime (k/sec): (4508847/3544)
 IV size: 16 bytes

R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 1
 encryption algorithm: AES - Advanced Encryption Standard (128 bit keys).
 hash algorithm: Secure Hash Standard
 authentication method: Pre-Shared Key
 Diffie-Hellman group: #5 (1536 bit)
 lifetime: 86400 seconds, no volume limit
 replay detection support: Y
 Status: ACTIVE
 inbound ah sas:
 inbound pcp sas:
 outbound esp sas:
 spi: 0x4F33CBD1(1328794577)
 transform: esp-aes esp-sha-hmac ,
 in use settings ={Tunnel, }
 conn id: 2002, flow_id: SW:2, crypto map: CMAP
 sa timing: remaining key lifetime (k/sec): (4508847/3536)
 IV size: 16 bytes
 replay detection support: Y
 Status: ACTIVE
 outbound ah sas:
 outbound pcp sas:
```