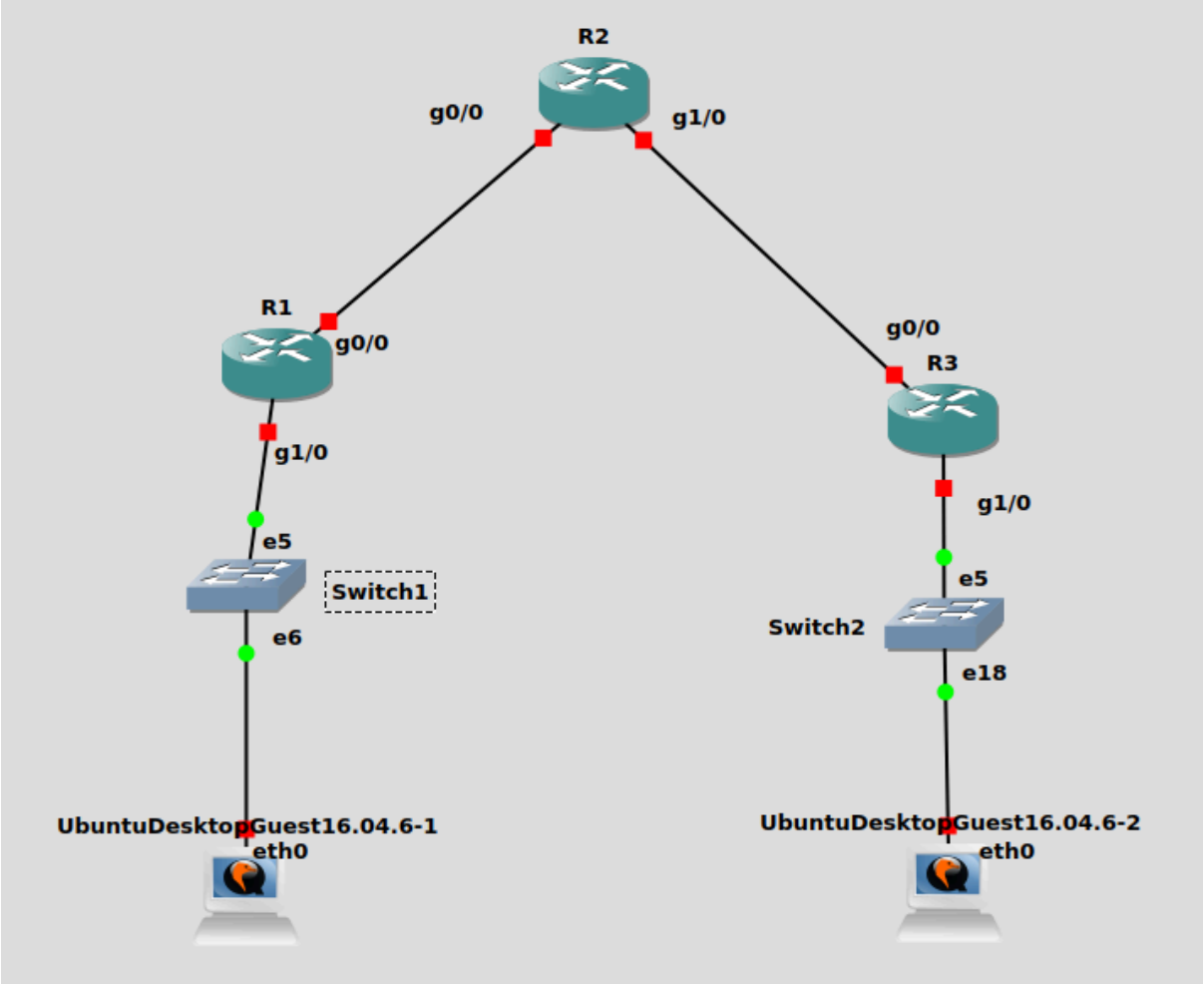
FOFANA Abdel-malik
FOURMONT Thibault

Configure a Site-to-Site VPN Topology

Part 1: Configure Basic Device Settings:

Step 1: Cable the network as shown in the topology:

On a décidé de choisir GNS3 pour notre topology



Step 2: Configure basic settings for each router.

On configure le nom des routeurs ainsi que les interfaces
avec les commandes :

hostname R1

interface GigabitEthernet0/0
ip address 10.1.1.1 255.255.255.252
no shutdown
exit
interface GigabitEthernet1/0
ip address 192.168.1.1 255.255.255.0
no shutdown

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            unassigned      YES unset  administratively down down
GigabitEthernet0/0     10.1.1.1        YES manual up                    up
GigabitEthernet1/0     192.168.1.1     YES manual up                    up
R1#
```

On peut voir que tout est ok
On fait pareil pour R2 et R3 mais en changeant les ip des interfaces

Step 3: Disable DNS lookup.

Cette étape est importante pour éviter que le routeur tente de résoudre des commandes mal saisies en adresses DNS, ce qui peut ralentir votre configuration.
Voici la commande qu'on a fait sur tous les routeurs
no ip domain lookup

```
R3(config)#no ip domain lookup
R3(config)#
```

Step 4: Configure the OSPF routing protocol on R1, R2, and R3.

En suite on configure OSPF (Open Shortest Path First) pour permettre aux routeurs de partager les informations de routage dynamiquement.

Voici les commandes que l'on a fait pour R1 ,R2 et R3
R1 :
router ospf 101
network 192.168.1.0 0.0.0.255 area 0
network 10.1.1.0 0.0.0.3 area 0
R2:
router ospf 101
network 10.1.1.0 0.0.0.3 area 0
network 10.2.2.0 0.0.0.3 area 0
R3:
router ospf 101
network 192.168.3.0 0.0.0.255 area 0
network 10.2.2.0 0.0.0.3 area 0

On a bien fait attention au masque (pour 255.255.255.0 et 255.255.255.252)

tous les routeurs dans la même zone (zone 0) devraient commencer à échanger leurs informations de routage. Pour vérifier que tout fonctionne correctement,on fait la commande sur R2 par exemple

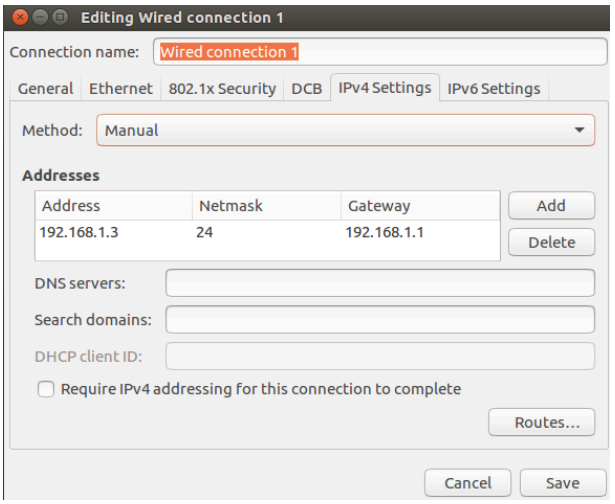
R1# show ip ospf neighbor

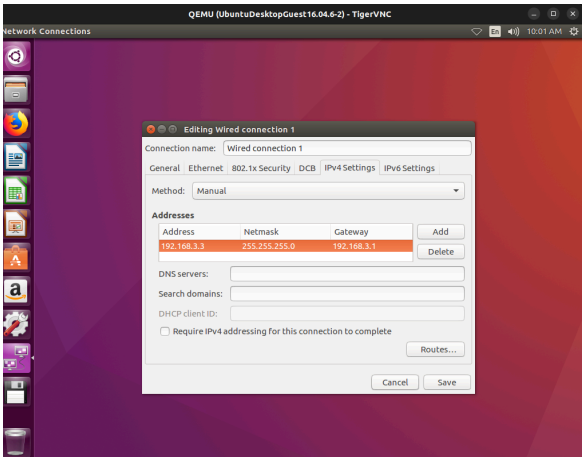
R2# show ip route

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.3.1       1   FULL/DR         00:00:30    10.2.2.1        GigabitEthernet1/0
192.168.1.1       1   FULL/DR         00:00:31    10.1.1.1        GigabitEthernet0/0
R2#
*Dec 26 15:26:43.631: %SYS-5-CONFIG_I: Configured from console by console
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/30 is directly connected, GigabitEthernet0/0
L       10.1.1.2/32 is directly connected, GigabitEthernet0/0
C       10.2.2.0/30 is directly connected, GigabitEthernet1/0
L       10.2.2.2/32 is directly connected, GigabitEthernet1/0
O     192.168.1.0/24 [110/2] via 10.1.1.1, 00:09:11, GigabitEthernet0/0
O     192.168.3.0/24 [110/2] via 10.2.2.1, 00:09:01, GigabitEthernet1/0
R2#
```

Step 5: Configure PC host IP settings.

Voici la configuration graphique que l'on a fait sur le pcA et le pc B





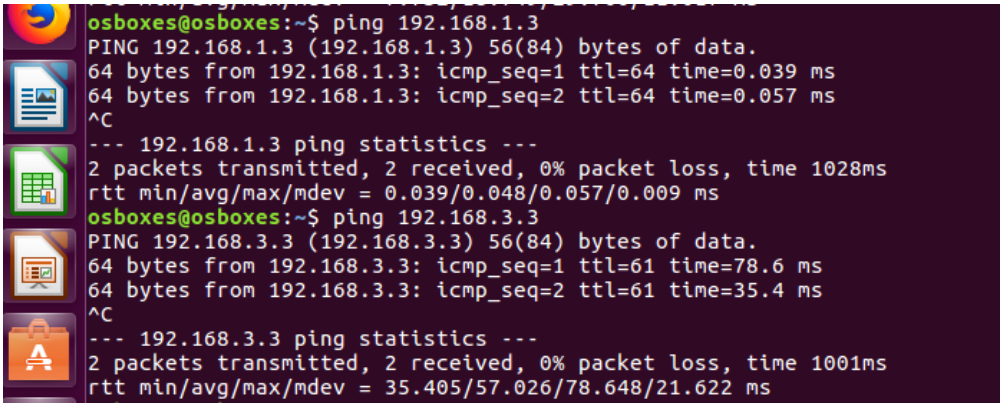
Step 6: Verify basic network connectivity.

a) Ping from R1 to the R3

```
R1# ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
R1#
```

Ca fonctionne Depuis **R1**, on peut **pinguer l'interface G0/0/1 de R3**

b) ping de PCA a PCB Les ip sont maintenant bien configuré et le pcA peut ping le PCB



Step 7: Configure and encrypt passwords.

On a fait ces commandes dans R1 et R3:
security passwords min-length 10
enable algorithm-type scrypt secret cisco12345
username admin01 algorithm-type scrypt secret admin01pass
On peut voir que tout fonctionne avec les commandes
show running-config | include username
show running-config | include enable secret

```
R1#show running-config | include username
username admin01 secret 9 $9$/IOJ3ZH3qLG9HU$6c.ClZ.kkI1Tx9JN39ndR2YpEzlaOOEmlEr.
drJt4WQ
R1#show running-config | include enable secret
enable secret 9 $9$n0qdBm99fnNJXk$lMrvXL7XEo0jPzH7iWGt/8zKNs5GaSy.6dseQfQMpXc
R1#
```

Step 8: Configure the console line.

line console 0
login local
exec-timeout 5 0
logging synchronous
Ces commandes configurent la ligne console pour exiger une connexion authentifiée, limiter l'inactivité à 5 minutes, et éviter l'interruption par les messages système. On fait les commandes pour show running-config pour verifier et on voit que tout est bon

```
R3(config)#line console 0
R3(config-line)#login local
R3(config-line)#exec-timeout 5 0
R3(config-line)#logging synchronous
R3(config-line)#
```

```
line con 0
 exec-timeout 5 0
 privilege level 15
 logging synchronous
 login local
 stopbits 1
```

Et après un 5 min d'inactivité il faut mettre le login mot de passe

```
*Dec 26 18:32:09.367: %SYS-5-CONFIG_I: Configured from console by console

User Access Verification

Username:
```

Step 9: Configure SSH Server.

Ensuite on fait ces commandes dans R1 et R3:

ip domain-name netsec.com
crypto key generate rsa general-keys modulus 2048
ip ssh version 2
line vty 0 4
login local
exec-timeout 5 0
transport input ssh

Voici un exemple ici avec R3

```
R3(config)#ip domain-name netsec.com
R3(config)#crypto key generate rsa general-keys modulus 2048
The name for the keys will be: R3.netsec.com

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 7 seconds)

R3(config)#ip ssh version 2
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#exec-timeout 5 0
R3(config-line)#transport input ssh
*Dec 26 21:51:51.299: %SSH-5-ENABLED: SSH 1.99 has been enabled
R3(config-line)#
```

Après l'étape 9, les routeurs sont configurés pour autoriser uniquement les connexions SSH sécurisées avec authentification locale, en utilisant des clés RSA pour le chiffrement et un délai de déconnexion de 5 minutes pour les sessions inactives.

Step 10: Save the basic running configuration for all three routers.

Et on sauvegarde la configuration de nos routeur avec la commande
copy running-config startup-config

```
R1#copy running-config startup-config
Destination filename [startup-config]?
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
[OK]
R1#
```

on peut se connect en ssh depuis le pcA par exemple

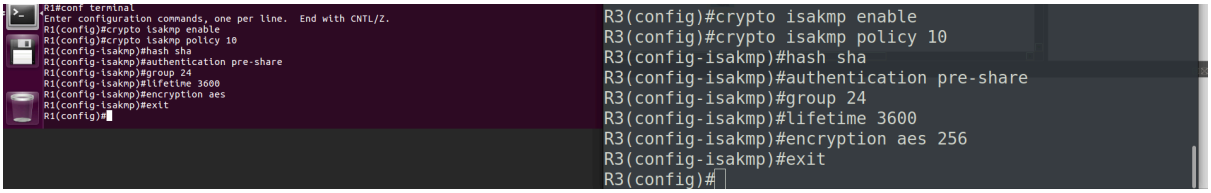
```
osboxes@osboxes:~/.ssh$ ssh admin01@192.168.1.1
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.
RSA key fingerprint is SHA256:O8mXE9S4nW+q/HYwimShExaGMvKR116ab99M5zgnXNk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.1' (RSA) to the list of known hosts.
Password:

R1>
```

Part 2: Configure a Site-to-Site VPN with Cisco IOS

Step 1 et Step 2: Enable IKE policies on R1 and R3. et Configure the IKE Phase 1 ISAKMP policy on R1 and R3.

Voici les commandes que l'on a fait sur R1 et R3

```
crypto isakmp enable
crypto isakmp policy 10
hash sha
authentication pre-share
group 24
lifetime 3600
encryption aes 256
exit
```



Ces commandes configurent et activent les politiques IKE Phase 1 pour l'IPsec VPN, en spécifiant que les échanges de clés utiliseront un chiffrement AES-256 pour la confidentialité, une fonction de hachage SHA pour l'intégrité des données, une authentification par clé pré-partagée, un groupe Diffie-Hellman 24 pour l'accord de clés sécurisé, et une durée de vie des associations de sécurité de 3600 secondes (1 heure),

avec la commande `crypto isakmp enable` permettant l'activation générale d'IKE sur le routeur.

```
R3#show crypto isakmp policy

Global IKE policy
Protection suite of priority 10
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #24 (2048 bit, 256 bit subgroup)
        lifetime:               3600 seconds, no volume limit
R3#
```

On peut voir avec la commande show crypto isakmp policy que tout est bon

Step 3: Configure pre-shared keys.

On configure les clé partagé pour le pair distant avec son ip (on fait ca pour R1 et R3)
R1: crypto isakmp key cisco123 address 10.2.2.1

R3 : crypto isakmp key cisco123 address 10.1.1.1

```
R3(config)#crypto isakmp key cisco123 address 10.1.1.1
```

Step 4: Configure the IPsec transform set and lifetime.

Pour comprendre a quel parametre sont disponible on fait la commande

crypto ipsec transform-set R1-R3 ?

```
R3(config)#crypto ipsec transform-set R1-R3 ?
  ah-md5-hmac      AH-HMAC-MD5 transform
  ah-sha-hmac      AH-HMAC-SHA transform
  ah-sha256-hmac   AH-HMAC-SHA256 transform
  ah-sha384-hmac   AH-HMAC-SHA384 transform
  ah-sha512-hmac   AH-HMAC-SHA512 transform
  comp-lzs         IP Compression using the LZS compression algorithm
  esp-3des         ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes          ESP transform using AES cipher
  esp-des          ESP transform using DES cipher (56 bits)
  esp-gcm          ESP transform using GCM cipher
  esp-gmac         ESP transform using GMAC cipher
  esp-md5-hmac     ESP transform using HMAC-MD5 auth
  esp-null         ESP transform w/o cipher
  esp-seal         ESP transform using SEAL cipher (160 bits)
  esp-sha-hmac     ESP transform using HMAC-SHA auth
  esp-sha256-hmac  ESP transform using HMAC-SHA256 auth
  esp-sha384-hmac  ESP transform using HMAC-SHA384 auth
  esp-sha512-hmac  ESP transform using HMAC-SHA512 auth
```

On configure un transform set et on y met une durée de vie dans R1 et R3 avec la commande :
crypto ipsec transform-set R1-R3 esp-aes 256 esp-sha-hmac

```
R3(config)#crypto ipsec transform-set R1-R3 esp-aes 256 esp-sha-hmac
R3(cfg-crypto-trans)#exit
```

Le transform set IPsec définit les algorithmes de chiffrement, d'authentification et d'encapsulation utilisés pour protéger et sécuriser les données transmises à travers le tunnel IPsec.
et on peut voir que tout est bon avec show crypto ipsec transform-set

```
R3#show crypto ipsec transform-set
Transform set default: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },
Transform set R1-R3: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },
```

et on fait la commande dans R1 et R3 crypto ipsec security-association lifetime seconds 1800
définit la durée de vie des associations de sécurité IPsec à 1800 secondes (30 minutes), après quoi une nouvelle négociation doit être initiée pour maintenir le tunnel sécurisé.

Step 5: Define interesting traffic.
On crée une ACL pour le traffic entre R1 et R3 dans R1
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
et vice versa dans R3
access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255

```
R1(config)#access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.2$
```

Ces ACLs indiquent aux routeurs quels paquets doivent être encryptés dans le tunnel IPsec. Tout paquet ne correspondant pas à ces règles sera transmis sans encryption. Une configuration miroir garantit que les deux côtés du VPN comprennent quels types de trafic sont sécurisés.

Step 6: Create and apply a crypto map.

L'objectif de cette étape est de créer et d'associer une crypto map à une interface pour chiffrer le trafic IPsec entre les routeurs.
on a fait sur R1 et R2 (en adaptant les commandes) les commandes suivantes:
crypto map CMAP 10 ipsec-isakmp
match address 101
set peer 10.2.2.1
set transform-set R1-R3
set pfs group24
set security-association lifetime seconds 900
exit
Cette configuration crée une crypto map nommée **CMAP** avec une priorité de 10, qui associe le trafic défini par l'ACL 101 à l'IP du pair distant (10.2.2.1), utilise le transform set **R1-R3** pour le chiffrement, active le PFS avec le groupe Diffie-Hellman 24, et définit une durée de vie de 900 secondes pour les associations de sécurité IPsec.

```
R1(config)#crypto map CMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R1(config-crypto-map)#match address 101
R1(config-crypto-map)#set peer 10.2.2.1
R1(config-crypto-map)#set transform-set R1-R3
R1(config-crypto-map)#set pfs group24
R1(config-crypto-map)#set security-association lifetime seconds 900
R1(config-crypto-map)#exit
R1(config)#
```

Application des crypto maps aux interfaces correctes (dans R1 et R3 aussi) avec les commandes :
interface GigabitEthernet0/0
crypto map CMAP

```
R1(config)#interface GigabitEthernet0/0
R1(config-if)#crypto map CMAP
R1(config-if)#exit
```

La crypto map est appliquée sur les interfaces **GigabitEthernet0/0** de R1 et R3, permettant de sécuriser le trafic défini comme intéressant via IPsec. Une fois le trafic détecté, les associations de sécurité (SA) seront négociées.

Step 7: Verify the IPsec configuration on R1 and R3.

L'objectif de cette étape est de valider la configuration des politiques et des crypto maps pour s'assurer que tout est correctement configuré avant d'établir le tunnel IPsec.
on peut voir avec les commandes
show crypto ipsec transform-set et show crypto map confirment que :

- Les transform sets définissent correctement les paramètres de chiffrement et d'authentification.
- Les crypto maps associent les ACL, les peers et les transform sets aux interfaces appropriées

```
R3#show crypto ipsec transform-set
Transform set default: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set R1-R3: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },
```

```
R3#show crypto map
Crypto Map IPv4 "CMAP" 10 ipsec-isakmp
        Peer = 10.1.1.1
        Extended IP access list 101
            access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
        Current peer: 10.1.1.1
        Security association lifetime: 4608000 kilobytes/900 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): Y
        DH group:  group24
        Mixed-mode : Disabled
        Transform sets={
                R1-R3:  { esp-256-aes esp-sha-hmac  } ,
        }
        Interfaces using crypto map CMAP:
                GigabitEthernet0/0
```

(pour R1)

```
R1#show crypto ipsec transform-set
Transform set default: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set R1-R3: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },

R1#show crypto map
Crypto Map IPv4 "CMAP" 10 ipsec-isakmp
        Peer = 10.2.2.1
        Extended IP access list 101
            access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
        Current peer: 10.2.2.1
        Security association lifetime: 4608000 kilobytes/900 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): Y
        DH group:  group24
        Mixed-mode : Disabled
        Transform sets={
                R1-R3:  { esp-256-aes esp-sha-hmac  } ,
        }
        Interfaces using crypto map CMAP:
                GigabitEthernet0/0
```

Step 8: Display ISAKMP security associations.

On peut voir avec la commande show crypto isakmp que aucun IKA SE existe

```
R1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status

IPv6 Crypto ISAKMP SA
```

Step 9: Display IPsec security associations.

```
R1#show crypto ipsec sa

interface: GigabitEthernet0/0
    Crypto map tag: CMAP, local addr 10.1.1.1

   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): (192.168.3.0/255.255.255.0/0/0)
   current_peer 10.2.2.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.1.1.1, remote crypto endpt.: 10.2.2.1
     plaintext mtu 1500, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet0/0
     current outbound spi: 0x0(0)
     PFS (Y/N): N, DH group: none

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
```

```
R3#show crypto ipsec sa

interface: GigabitEthernet0/0
    Crypto map tag: CMAP, local addr 10.2.2.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.3.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer 10.1.1.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.2.2.1, remote crypto endpt.: 10.1.1.1
     plaintext mtu 1500, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet0/0
     current outbound spi: 0x0(0)
     PFS (Y/N): N, DH group: none

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
```

Quand on fait la commande show crypto ipsec sa pour vérifier les associations de sécurité

Step 10: Generate some uninteresting test traffic and observe the results.

Dans R1 on a fait les commandes suivantes:

ping 10.2.2.1
show crypto isakmp sa
ping 192.168.3.1
show crypto isakmp sa
debug ip ospf hello
no debug ip ospf hello

```
Password:
R1#ping 10.2.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/24/40 ms
R1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status

IPv6 Crypto ISAKMP SA

R1#ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
R1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status

IPv6 Crypto ISAKMP SA

R1#debug ip ospf hello
OSPF hello debugging is on
R1#
*Dec 27 20:38:04.217: OSPF-101 HELLO Gi1/0: Send hello to 224.0.0.5 area 0 from 192.168.1.1
R1#no debug ip ospf hello
OSPF hello debugging is off
R1#
```

Réponses aux questions :

1. **Pourquoi aucune SA n'a été créée pour les pings ?**
    - **Réponse :** Les SA ne sont pas créées car les pings envoyés de R1 à R3 (10.2.2.1 ou 192.168.3.1) n'appartiennent pas au trafic défini comme **intéressant** par l'ACL 101. L'ACL 101 spécifie que le trafic intéressant provient du réseau 192.168.1.0/24 (LAN de R1) et est destiné au réseau 192.168.3.0/24 (LAN de R3).
2. **Pourquoi aucune SA n'a été créée pour le trafic OSPF ?**
    - **Réponse :** Le trafic OSPF est du trafic de routage interne entre routeurs (multidiffusion ou point à point) et n'est pas considéré comme **intéressant**. Par conséquent, il ne déclenche pas l'établissement d'une SA et n'est pas chiffré.

Step 11: Generate some interesting test traffic and observe the results.

Cette fois ci on fait le ping comme ca ce qui est plus interessant
ping 192.168.3.1 source 192.168.1.1Cette fois ci on fait le ping comme ca ce qui est plus interessant
ping 192.168.3.1 source 192.168.1.1
show crypto isakmp sa
Explication : Ce ping utilise une adresse source spécifique (192.168.1.1), qui correspond au trafic intéressant défini par l'ACL 101.
Résultat attendu : Le ping réussit et déclenche la création d'une association de sécurité (SA).

```
R1#ping 192.168.3.1 source 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1

*Dec 27 20:42:22.929: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Informational mode failed with peer at 10.2.2.1.....
Success rate is 0 percent (0/5)
R1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
10.2.2.1        10.1.1.1        MM_NO_STATE          0 ACTIVE

IPv6 Crypto ISAKMP SA

R1#
```

Réponses aux Questions :

1. **Pourquoi une SA a-t-elle été créée cette fois-ci ?**
    - **Réponse :** L'adresse source du trafic (192.168.1.1) et l'adresse de destination (192.168.3.1) correspondent au trafic intéressant défini dans l'ACL 101. Cela a déclenché la négociation IPsec et l'établissement d'une SA.
2. **Quels sont les points d'extrémité du tunnel IPsec VPN ?**
    - **Réponse :** Source : 10.1.1.1 (interface G0/0 de R1), Destination : 10.2.2.1 (interface G0/0 de R3).

```
osboxes@osboxes:~$ ping 192.168.3.3
PING 192.168.3.3 (192.168.3.3) 56(84) bytes of data.
64 bytes from 192.168.3.3: icmp_seq=3 ttl=62 time=38.8 ms
64 bytes from 192.168.3.3: icmp_seq=4 ttl=62 time=46.3 ms
64 bytes from 192.168.3.3: icmp_seq=5 ttl=62 time=35.3 ms
64 bytes from 192.168.3.3: icmp_seq=6 ttl=62 time=32.1 ms
64 bytes from 192.168.3.3: icmp_seq=7 ttl=62 time=38.8 ms
64 bytes from 192.168.3.3: icmp_seq=8 ttl=62 time=34.7 ms
^C
--- 192.168.3.3 ping statistics ---
8 packets transmitted, 6 received, 25% packet loss, time 7033ms
rtt min/avg/max/mdev = 32.183/37.723/46.319/4.497 ms
osboxes@osboxes:~$
```

```
R1#show crypto ipsec sa

interface: GigabitEthernet0/0
    Crypto map tag: CMAP, local addr 10.1.1.1

   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): (192.168.3.0/255.255.255.0/0/0)
   current_peer 10.2.2.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 6, #pkts encrypt: 6, #pkts digest: 6
    #pkts decaps: 6, #pkts decrypt: 6, #pkts verify: 6
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.1.1.1, remote crypto endpt.: 10.2.2.1
     plaintext mtu 1342, path mtu 1400, ip mtu 1400, ip mtu idb GigabitEthernet0/0
     current outbound spi: 0x3480A901(880847105)
     PFS (Y/N): Y, DH group: group24

     inbound esp sas:
 --More--
```

Combien de paquets ont été transformés entre R1 et R3 ?

- **Réponse :** Le nombre de paquets varie en fonction des pings envoyés. Par exemple, dans les résultats donnés, **6 paquets encapsulés** et **6 paquets décapsulés** sont affichés.

Quels autres types de trafic peuvent déclencher la création d'une SA ?

- **Réponse :** Tout trafic correspondant à l'ACL 101, comme FTP, HTTP, Telnet, ou tout autre protocole, entre les réseaux **192.168.1.0/24** et **192.168.3.0/24**.

Vos résultats montrent que le tunnel IPsec est fonctionnel avec les statistiques adéquates. Si les pertes persistent, vérifiez la MTU comme décrit précédemment, mais la configuration actuelle est correcte et fonctionnelle.