

29/12/2024

Protocole PPP et VPN

Eloham Caron
BTS SIO 2 SISR

Table des matières

1. Introduction.....	2
a) PPP (Point-to-Point Protocol) :.....	2
b) CHAP (Challenge-Handshake Authentication Protocol) :	2
2. Configuration d'une liaison avec les protocoles PPP et CHAP	3
c) Vérifiez le bon fonctionnement de la liaison avec l'authentification CHAP pour l'encapsulation PPP.	4
d) Quelle(s) différence(s) peut-on observer avec une authentification PAP ?	5
e) CHAP et PPP : son complémentair.....	5
3. Création d'un tunnel VPN IP Sec entre 2 routeurs avec Packet Tracer	6
Configuration OSPF	6
f) Verification OSPF	8
g) Configuration du VPN IPSEC.....	8
h) Configuration VPN IPsec : Explications des choix.....	10
i) Vérification des paquets cryptés	11
j) Vérification des paquets IPsec sur Valence.....	12
k) Vérifications avec ping et encapsulation	13
4. VPN Site à site, application.....	15
l) Activez ISAKMP (IKE - Internet Key Exchange) :	17
m) Définition de la clé d'échange VPNKEY	17
n) Configuration des méthodes de cryptage et d'authentification :	18
o) Modification des ACL pour le NAT :	18
p) Problème des ACL sur Packet Tracer :	19
q) Solution : Remplacer l'ACL pour forcer sa mise à jour	19
5. Configuration du routeur de QAQORTOQ.....	20
r) Configuration de l'ACL LAN (celle du NAT)	21
s) Tests et vérifications.....	21
6. Proposition de solution en cas de dysfonctionnement :	22
7. routes	23

1. INTRODUCTION

Ce TP se concentre sur la mise en place d'un VPN (Virtual Private Network), une solution essentielle pour sécuriser les échanges entre deux sites distants ou entre des utilisateurs et leur réseau d'entreprise. Un VPN crée un "tunnel" sécurisé au sein d'un réseau public, comme Internet, permettant de transmettre des données de manière confidentielle et fiable. Ces technologies sont largement utilisées dans les entreprises pour connecter des filiales, permettre le télétravail ou protéger les données sensibles.

Dans ce TP, nous allons explorer différentes couches de sécurité et de connectivité nécessaires à un VPN. PPP (Point-to-Point Protocol) sera utilisé pour établir une liaison point à point standardisée, essentielle pour la communication de bout en bout. Pour renforcer cette liaison, nous configurerons CHAP (Challenge Handshake Authentication Protocol), qui ajoute une couche d'authentification sécurisée grâce à un mécanisme de challenge-réponse.

Enfin, le cœur de ce TP portera sur IPsec (Internet Protocol Security), une technologie incontournable pour les VPN modernes. IPsec permettra de chiffrer les données transmises, de garantir leur intégrité et d'assurer qu'elles ne soient ni interceptées ni modifiées par des tiers.

En abordant ces aspects, ce TP permettra non seulement de comprendre comment configurer un VPN, mais aussi de saisir l'importance de chaque composant dans la sécurité et la fiabilité des réseaux d'entreprise. Vous apprendrez à mettre en place une solution complète, adaptée aux besoins des entreprises modernes pour sécuriser leurs communications.

a) PPP (POINT-TO-POINT PROTOCOL) :

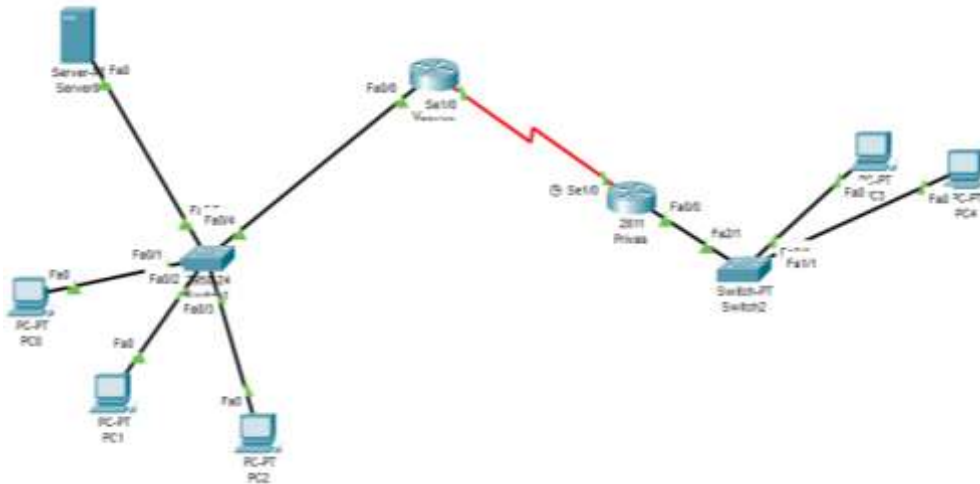
PPP est un protocole de communication standard défini dans la **RFC 1661** publiée en **juillet 1994**. Il est utilisé pour établir une connexion directe entre deux nœuds réseau, généralement sur des liaisons série ou des connexions modem. PPP prend en charge plusieurs protocoles de couche réseau (comme IPv4 et IPv6) et offre des mécanismes pour l'encapsulation, la négociation des paramètres de connexion (via LCP), et l'authentification. Grâce à ses sous-protocoles comme PAP et CHAP, PPP garantit une compatibilité multi-vendeurs et une flexibilité dans les configurations réseau.

b) CHAP (CHALLENGE-HANDSHAKE AUTHENTICATION PROTOCOL) :

CHAP est un protocole d'authentification basé sur un échange sécurisé, défini initialement dans la **RFC 1334** publiée en **octobre 1992**. CHAP fonctionne en trois étapes : le serveur envoie un "challenge" au client, le client répond avec une réponse cryptée basée sur un mot de passe partagé, et le serveur valide cette réponse. Contrairement à PAP, CHAP offre une sécurité accrue en évitant de transmettre les mots de passe en clair et en renouvelant périodiquement les challenges. Ce mécanisme est souvent utilisé conjointement avec PPP pour garantir la sécurité des connexions point à point.

2. CONFIGURATION D'UNE LIAISON AVEC LES PROTOCOLES PPP ET CHAP

Dans cette configuration réseau, nous allons mettre en place le protocole **PPP** avec l'authentification **CHAP** sur la liaison série entre les deux routeurs (2811 Valence et 2811 Privas). Cette implémentation permettra d'assurer une connexion sécurisée et fiable entre les deux routeurs. Une fois configuré, PPP avec CHAP vérifiera l'identité des routeurs en utilisant un mécanisme de challenge-réponse, renforçant ainsi la sécurité sur cette liaison critique.



Configuration du CHAP :

Valence
<pre>hostname Valence username Privas password 0 cisco123 interface Serial1/2 no ip address clock rate 2000000 shutdown router ospf 100 log-adjacency-changes network 192.168.1.0 0.0.0.255 area 10 network 200.1.1.0 0.0.0.255 area 10 interface Serial1/0 ip address 200.1.1.1 255.255.255.0 encapsulation ppp ppp authentication chap</pre>

Fonctionnement rapide de la configuration :

- Les commandes hostname et username définissent les identifiants pour CHAP.
- Les interfaces série utilisent l'encapsulation PPP avec l'option ppp authentication chap.
- Les réseaux OSPF sont définis pour permettre le partage des routes dynamiques via la liaison sécurisée.
- Ainsi, cette configuration assure à la fois la connectivité et la sécurité sur la liaison série tout en facilitant le routage dynamique avec OSPF.

Privas
<pre>hostname Privas username Valence password 0 cisco123</pre>

```
interface Serial1/2
no ip address
clock rate 2000000
shutdown
router ospf 100
log-adjacency-changes
network 192.168.2.0 0.0.0.255 area 10
network 200.1.1.0 0.0.0.255 area 10

interface Serial1/0
ip address 200.1.1.2 255.255.255.0
encapsulation ppp
ppp authentication chap
```

Pourquoi cette configuration ?

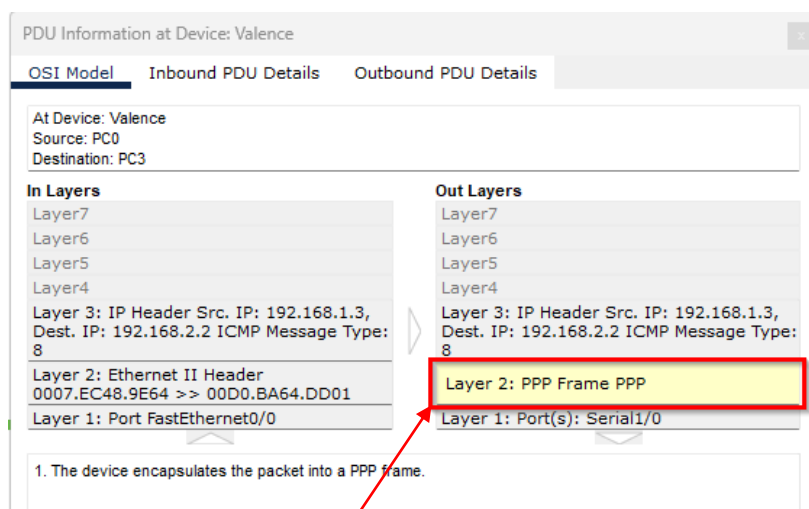
Sécurité renforcée : CHAP évite d'envoyer des mots de passe en clair sur le réseau, réduisant ainsi le risque d'interception.

OSPF avec PPP : Les réseaux OSPF configurés ici (areas 10) nécessitent une liaison fiable et sécurisée entre les routeurs pour échanger les informations de routage.

c) VERIFIEZ LE BON FONCTIONNEMENT DE LA LIAISON AVEC L'AUTHENTIFICATION CHAP POUR L'ENCAPSULATION PPP.

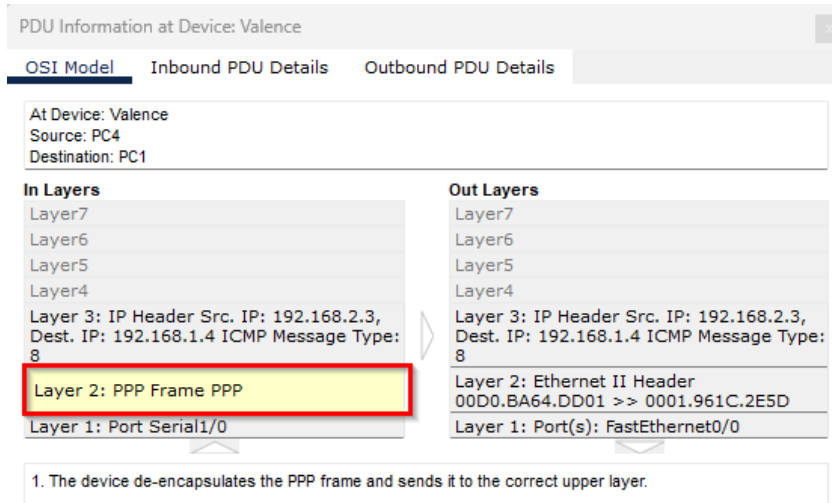
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC3	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC4	PC1	ICMP		0.000	N	1	(edit)	(delete)

Dans cette capture, on peut voir que les tests ICMP (pings) envoyés entre différents PC (PC0 vers PC3 et PC4 vers PC1) sont réussis. Cela confirme que la communication entre ces équipements à travers le réseau est fonctionnelle. Le succès des pings prouve que la liaison série entre les routeurs est correctement configurée avec l'encapsulation PPP et l'authentification CHAP. Ce test est essentiel pour valider que le réseau est opérationnel avant de déployer d'autres services ou configurations.



Le paquet est encapsulé en trame **PPP** au niveau de la couche 2 pour traverser l'interface série (**Serial1/0**). Cela montre que l'encapsulation PPP fonctionne correctement, ce qui est

une exigence pour permettre la liaison point à point avec authentification CHAP. Ce mécanisme assure une compatibilité avec les autres protocoles et renforce la sécurité de la transmission.



On observe le chemin inverse, où un paquet envoyé depuis PC4 à destination de PC1 est décapsulé sur l'interface série (Serial1/0) du routeur Valence. La trame PPP est transformée en trame Ethernet II, ce qui permet de router le paquet vers la bonne destination au sein du réseau local. Ce processus de décapsulation valide le fonctionnement de l'encapsulation PPP et confirme que la liaison est entièrement opérationnelle avec l'authentification CHAP.

d) QUELLE(S) DIFFERENCE(S) PEUT-ON OBSERVER AVEC UNE AUTHENTIFICATION PAP ?

L'authentification PAP (Password Authentication Protocol) présente des différences notables par rapport à CHAP (Challenge-Handshake Authentication Protocol), notamment en matière de sécurité et de fonctionnement. Avec PAP, le mot de passe est transmis en clair sur le réseau, ce qui le rend vulnérable aux attaques d'interception. En revanche, CHAP utilise une méthode de défi-réponse, garantissant que le mot de passe n'est jamais transmis directement. CHAP procède à une authentification périodique et mutuelle, ce qui signifie que le serveur peut solliciter plusieurs vérifications pendant la connexion pour prévenir les risques d'usurpation. Cette approche offre une meilleure protection contre les attaques par rejeu et le sniffing. Par conséquent, CHAP est beaucoup plus sécurisé que PAP, qui reste simple mais adapté uniquement à des environnements où la sécurité n'est pas une priorité.

e) CHAP ET PPP : SON COMPLEMENTAIRE

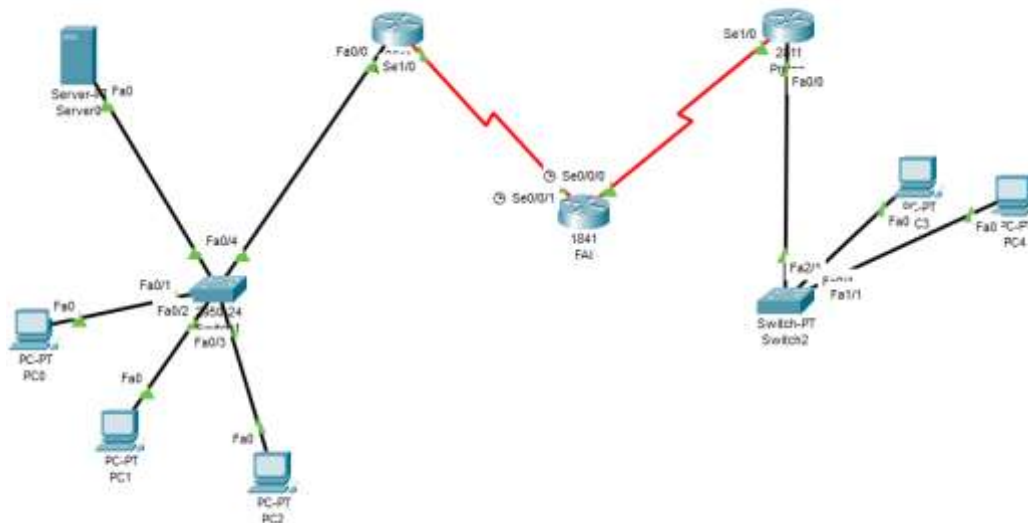
Le protocole **PPP (Point-to-Point Protocol)** est conçu pour établir une connexion point à point fiable entre deux équipements, notamment sur des liaisons série. Il offre une encapsulation des données pour la transmission et des mécanismes de négociation, mais ne gère pas directement la sécurité de l'authentification.

C'est là qu'intervient **CHAP (Challenge Handshake Authentication Protocol)**. CHAP est un protocole d'authentification qui complète PPP en ajoutant une couche de sécurité. Alors que PPP assure le transport des données, CHAP garantit que les deux parties de la connexion sont bien authentifiées. CHAP utilise un mécanisme de challenge-réponse et des mots de passe chiffrés pour éviter les transmissions non sécurisées.

En résumé, PPP et CHAP sont complémentaires : PPP établit et gère la connexion point à point, tandis que CHAP sécurise cette connexion en vérifiant l'identité des équipements, rendant l'ensemble fiable et adapté à des environnements sensibles.

3. CREATION D'UN TUNNEL VPN IP SEC ENTRE 2 ROUTEURS AVEC PACKET TRACER

Dans cette partie, nous allons mettre en place ces éléments en configurant successivement l'authentification CHAP, l'encapsulation PPP et le protocole OSPF, avant de passer à la configuration du tunnel VPN IPsec. Cette approche étape par étape permettra de vérifier la connectivité et la sécurité à chaque niveau.



CONFIGURATION OSPF

L'association d'OSPF et CHAP permet de garantir à la fois la connectivité et la sécurité. OSPF assure le routage dynamique en échangeant automatiquement les informations sur les réseaux disponibles, ce qui optimise la communication entre sous-réseaux. CHAP, de son côté, sécurise la liaison série en authentifiant les routeurs, empêchant tout accès non autorisé. Ensemble, ils assurent un réseau fiable, sécurisé et adaptable aux changements.

Configuration du CHAP :

```
Valence

router ospf 1
 network 192.168.1.0 0.0.0.255 area 0
 network 200.1.2.0 0.0.0.255 area 0

interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0

interface Serial1/0
 ip address 200.1.2.1 255.255.255.0
```

Eloham Caron
PROTOCOLE PPP ET VPN

```
Valence>en
Valence#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Valence(config)#router ospf 1
Valence(config-router)# network 192.168.1.0 0.0.0.255 area 0
Valence(config-router)# network 200.1.2.0 0.0.0.255 area 0
Valence(config-router)#
Valence(config-router)#interface FastEthernet0/0
Valence(config-if)# ip address 192.168.1.1 255.255.255.0
Valence(config-if)#
Valence(config-if)#interface Serial1/0
Valence(config-if)# ip address 200.1.2.1 255.255.255.0
Valence(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
|
```

FAI

```
router ospf 1
 network 200.1.1.0 0.0.0.255 area 0
 network 200.1.2.0 0.0.0.255 area 0

interface Serial0/0/1
 ip address 200.1.2.254 255.255.255.0

interface Serial0/0/0
 ip address 200.1.1.254 255.255.255.0
```

```
FAI>en
FAI#conf t
Enter configuration commands, one per line. End with CNTL/Z.
FAI(config)#router ospf 1
FAI(config-router)# network 200.1.1.0 0.0.0.255 area 0
FAI(config-router)# network 200.1.2.0 0.0.0.255 area 0
FAI(config-router)#
FAI(config-router)#interface Serial0/0/1
FAI(config-if)# ip address 200.1.2.254 255.255.255.0
FAI(config-if)#
FAI(config-if)#interface Serial0/0/0
FAI(config-if)# ip address 200.1.1.254 255.255.255.0
FAI(config-if)#|
```

Privas

```
router ospf 1
 network 192.168.2.0 0.0.0.255 area 0
 network 200.1.1.0 0.0.0.255 area 0

interface FastEthernet0/0
 ip address 192.168.2.1 255.255.255.0

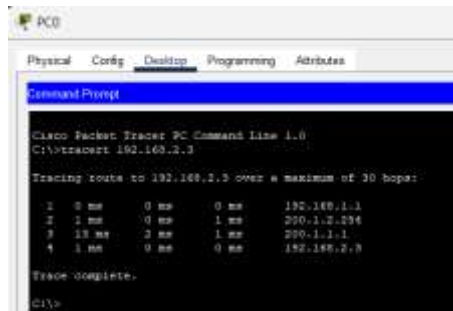
interface Serial1/0
 ip address 200.1.1.1 255.255.255.0
```

```
Privas>en
Privas#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Privas(config)#router ospf 1
Privas(config-router)# network 192.168.2.0 0.0.0.255 area 0
Privas(config-router)# network 200.1.1.0 0.0.0.255 area 0
Privas(config-router)#
Privas(config-router)#interface FastEthernet0/0
Privas(config-if)# ip address 192.168.2.1 255.255.255.0
Privas(config-if)#
Privas(config-if)#interface Serial1/0
Privas(config-if)# ip address 200.1.1.1 255.255.255.0
Privas(config-if)#
```


f) VERIFICATION OSPF

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC4	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC1	PC4	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC3	PC1	ICMP		0.000	N	3	(edit)	(delete)

Les connexions OSPF fonctionnent correctement. Chaque test ICMP (ping) entre les différentes machines (PC0, PC1, PC2, PC3, et PC4) a un statut "Successful", ce qui signifie que le routage OSPF est bien configuré et que les routes entre les différents réseaux sont apprises et fonctionnelles.



Le résultat de la commande **tracert 192.168.2.3** depuis le PC0 montre que le trafic traverse successivement les routeurs et les réseaux configurés via OSPF. Les différentes étapes confirment que les routes ont été **correctement propagées** et que chaque routeur connaît le chemin optimal pour atteindre la destination. Cette vérification prouve que le protocole OSPF est fonctionnel et que le routage dynamique entre les sous-réseaux est correctement établi.

g) CONFIGURATION DU VPN IPSEC

Valence
<pre>crypto isakmp enable crypto isakmp policy 10 authentication pre-share encryption 3des hash md5 group 5 lifetime 3600 crypto isakmp key 12345 address 200.1.1.1 crypto ipsec transform-set 50 esp-3des esp-md5-hmac crypto map tunnel-Privas 10 ipsec-isakmp set peer 200.1.1.1 set transform-set 50 set security-association lifetime seconds 900 access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 interface s1/0 crypto map tunnel-Privas</pre>

Eloham Caron
PROTOCOLE PPP ET VPN

```
Valence>en
Valence#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Valence(config)#crypto isakmp enable
Valence(config)#crypto isakmp policy 10
Valence(config-isakmp)# authentication pre-share
Valence(config-isakmp)# encryption 3des
Valence(config-isakmp)# hash md5
Valence(config-isakmp)# group 5
Valence(config-isakmp)# lifetime 3600
Valence(config-isakmp)#crypto isakmp key 12345 address 200.1.1.1
A pre-shared key for address mask 200.1.1.1 255.255.255.255 already exists!
Valence(config)#crypto ipsec transform-set 50 esp-3des esp-md5-hmac
Valence(config)#crypto map tunnel-Privas 10 ipsec-isakmp
Valence(config-crypto-map)# set peer 200.1.1.1
Valence(config-crypto-map)# set transform-set 50
Valence(config-crypto-map)# set security-association lifetime seconds 900
Valence(config-crypto-map)#access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Valence(config)#interface s1/0
Valence(config-if)# crypto map tunnel-Privas
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
Valence(config-if)#
Valence(config-if)#
```

Privas

```
crypto isakmp enable
crypto isakmp policy 10
 authentication pre-share
 encryption 3des
 hash md5
 group 5
 lifetime 3600
crypto isakmp key 12345 address 200.1.2.1
crypto ipsec transform-set 50 esp-3des esp-md5-hmac
crypto map tunnel-Valence 10 ipsec-isakmp
 set peer 200.1.2.1
 set transform-set 50
 set security-association lifetime seconds 900
access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
interface s1/0
 crypto map tunnel-Valence
```

```
Privas>en
Privas#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Privas(config)#crypto isakmp enable
Privas(config)#crypto isakmp policy 10
Privas(config-isakmp)# authentication pre-share
Privas(config-isakmp)# encryption 3des
Privas(config-isakmp)# hash md5
Privas(config-isakmp)# group 5
Privas(config-isakmp)# lifetime 3600
Privas(config-isakmp)#crypto isakmp key 12345 address 200.1.2.1
A pre-shared key for address mask 200.1.2.1 255.255.255.255 already exists!
Privas(config)#crypto ipsec transform-set 50 esp-3des esp-md5-hmac
Privas(config)#crypto map tunnel-Valence 10 ipsec-isakmp
Privas(config-crypto-map)# set peer 200.1.2.1
Privas(config-crypto-map)# set transform-set 50
Privas(config-crypto-map)# set security-association lifetime seconds 900
Privas(config-crypto-map)#access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Privas(config)#interface s1/0
Privas(config-if)# crypto map tunnel-Valence
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
Privas(config-if)#
```

h) CONFIGURATION VPN IPSEC : EXPLICATIONS DES CHOIX

Pour établir un tunnel VPN IPsec, la première étape consiste à activer **ISAKMP** avec la commande `crypto isakmp enable`. Cela permet de gérer automatiquement la négociation des paramètres de sécurité entre les deux routeurs.

Ensuite, une politique ISAKMP est définie avec `crypto isakmp policy 10`. Cette politique précise les paramètres de sécurité à utiliser pour la négociation, notamment :

- **Chiffrement 3DES** pour garantir la confidentialité des données,
- **MD5** pour vérifier l'intégrité des données échangées,
- Une clé Diffie-Hellman (groupe 5) pour un échange sécurisé des clés.

L'authentification est basée sur une clé pré-partagée (`crypto isakmp key`), qui simplifie la configuration tout en offrant une sécurité adaptée.

Pour chiffrer et authentifier les données IPsec, un **Transform Set** est configuré (`crypto ipsec transform-set 50 esp-3des esp-md5-hmac`). Cela précise comment les données seront protégées lorsqu'elles transitent par le tunnel.

La carte `crypto` (`crypto map tunnel-Privas`) lie ces configurations et associe l'adresse du routeur distant, le transform set et la durée de vie des associations de sécurité. Cette étape est cruciale pour définir précisément le trafic qui sera sécurisé par le VPN.

Enfin, la carte `crypto` est appliquée sur l'interface série (`interface s1/0 crypto map`), activant ainsi IPsec pour sécuriser le trafic. Une liste d'accès (`access-list 101`) garantit que seul le trafic entre les réseaux locaux (192.168.1.0 et 192.168.2.0) passe par le tunnel, offrant ainsi un contrôle précis sur les communications sécurisées.

Verification du VPN

Valence : `show crypto map`

La commande `show crypto map` sur le routeur Valence confirme que la carte `crypto tunnel-Privas` est bien configurée et appliquée à l'interface `Serial1/0`. Elle montre également que l'adresse IP du pair (200.1.1.1), la liste d'accès 101 et le transform set 50 sont correctement associés. Ces éléments valident que les paramètres IPsec sont prêts à sécuriser les échanges avec le réseau distant.

```
Valence#show crypto map
Crypto Map tunnel-Privas 10 ipsec-isakmp
  Peer = 200.1.1.1
  Extended IP access list 101
    access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
  Current peer: 200.1.1.1
  Security association lifetime: 4608000 kilobytes/900 seconds
  PFS (Y/N): N
  Transform sets={
    50,
  }
  Interfaces using crypto map tunnel-Privas:
    Serial1/0

Valence#show access-lists
Extended IP access list 101
  10 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 (1 match(es))
```

La commande `show access-lists` affiche la liste d'accès 101 sur Valence. Celle-ci autorise le trafic entre les réseaux 192.168.1.0 et 192.168.2.0. La présence d'un "match" indique que le trafic correspondant passe bien par le tunnel IPsec, confirmant ainsi que la configuration est fonctionnelle.

Privas : `show crypto map`

```
Privas#show crypto map
Crypto Map tunnel-Valence 10 ipsec-isakmp
  Peer = 200.1.2.1
  Extended IP access list 101
    access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
  Current peer: 200.1.2.1
  Security association lifetime: 4608000 kilobytes/900 seconds
  PFS (Y/N): N
  Transform sets={
    50,
  }
  Interfaces using crypto map tunnel-Valence:
    Serial1/0
```

Sur le routeur Privas, la commande `show crypto map` affiche des informations similaires pour la carte crypto tunnel-Valence. Elle confirme que l'adresse IP du pair (200.1.2.1), la liste d'accès 101 et le transform set 50 sont appliqués à l'interface Serial1/0, validant la symétrie de la configuration entre les deux routeurs.

```
Privas#show access-lists
Extended IP access list 101
  10 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
```

La commande `show access-lists` sur Privas montre la liste d'accès 101, qui autorise le trafic entre les réseaux 192.168.2.0 et 192.168.1.0. Ici aussi, la présence d'un "match" prouve que le trafic est acheminé correctement via le tunnel VPN.

i) VERIFICATION DES PAQUETS CRYPTES

Afin de vérifier si les paquets envoyés sont cryptés et que les paquets reçus sont bien décryptés, vous pouvez le vérifier avec la commande suivante :

Privas : `sh crypto ipsec sa`

La commande `show crypto ipsec sa` sur le routeur Privas affiche le statut des associations de sécurité (SA). On y voit les paquets encapsulés et chiffrés en sortie ainsi que les paquets déchiffrés en entrée. Ces statistiques confirment que le trafic passe bien par le tunnel IPsec et est protégé par les mécanismes de chiffrement et d'intégrité. Le statut "ACTIVE" valide que le tunnel est opérationnel.

```
Privas#sh crypto ipsec sa

interface: Serial1/0
  Crypto map tag: tunnel-Valence, local addr 200.1.1.1

protected vrf: (none)
local  ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
remote  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
current_peer 200.1.2.1 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 6, #pkts encrypt: 6, #pkts digest: 0
#pkts decaps: 6, #pkts decrypt: 6, #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.: 200.1.1.1, remote crypto endpt.:200.1.2.1
path mtu 1500, ip mtu 1500, ip mtu idb Serial1/0
current outbound spi: 0x0197EBE1(26733537)

inbound esp sas:
  spi: 0x86FDF56B(2264790379)
    transform: esp-3des esp-md5-hmac ,
    in use settings ={Tunnel, }
    conn id: 2009, flow_id: FPGA:1, crypto map: tunnel-Valence
    sa timing: remaining key lifetime (k/sec): (4525504/414)
    IV size: 16 bytes
    replay detection support: N
    Status: ACTIVE

inbound ah sas:

inbound pcsp sas:

outbound esp sas:
  spi: 0x0197EBE1(26733537)
    transform: esp-3des esp-md5-hmac ,
    in use settings ={Tunnel, }
    conn id: 2010, flow_id: FPGA:1, crypto map: tunnel-Valence
    sa timing: remaining key lifetime (k/sec): (4525504/414)
    IV size: 16 bytes
    replay detection support: N
    Status: ACTIVE

outbound ah sas:

outbound pcsp sas:

Privas#
```

j) VERIFICATION DES PAQUETS IPSEC SUR VALENCE

La commande `show crypto ipsec sa` sur le routeur Valence confirme que les paquets sont correctement encapsulés et chiffrés en sortie, puis déchiffrés en entrée. Les statistiques affichées (6 paquets encryptés/déchiffrés) valident que le tunnel IPsec est actif et que le trafic entre les réseaux 192.168.1.0 et 192.168.2.0 est sécurisé. Le statut "ACTIVE" pour les associations de sécurité (SA) montre que la configuration fonctionne comme prévu.

Eloham Caron PROTOCOLE PPP ET VPN

```
Valence#sh crypto ipsec sa

interface: Serial1/0
  Crypto map tag: tunnel-Privas, local addr 200.1.2.1

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 200.1.1.1 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 6, #pkts encrypt: 6, #pkts digest: 0
#pkts decaps: 6, #pkts decrypt: 6, #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 1, #recv errors 0

local crypto endpt.: 200.1.2.1, remote crypto endpt.:200.1.1.1
path mtu 1500, ip mtu 1500, ip mtu idb Serial1/0
current outbound spi: 0x86FDF56B(2264790379)

inbound esp sas:
  spi: 0x0197EBE1(26733537)
    transform: esp-3des esp-md5-hmac ,
    in use settings ={Tunnel, }
    conn id: 2009, flow_id: FPGA:1, crypto map: tunnel-Privas
    sa timing: remaining key lifetime (k/sec): (4525504/417)
    IV size: 16 bytes
    replay detection support: N
    Status: ACTIVE

inbound ah sas:

inbound pcip sas:

outbound esp sas:
  spi: 0x86FDF56B(2264790379)
    transform: esp-3des esp-md5-hmac ,
    in use settings ={Tunnel, }
    conn id: 2010, flow_id: FPGA:1, crypto map: tunnel-Privas
    sa timing: remaining key lifetime (k/sec): (4525504/417)
    IV size: 16 bytes
    replay detection support: N
    Status: ACTIVE

outbound ah sas:

outbound pcip sas:
```

k) VERIFICATIONS AVEC PING ET ENCAPSULATION

Pour vérifier la bonne fonctionnalité du tunnel VPN IPsec, deux étapes clés sont nécessaires :

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC0	ICMP
	0.000	--	PC0	ICMP
	0.001	PC0	Switch1	ICMP
	0.001	--	PC0	ICMP
	0.002	PC0	Switch1	ICMP
	0.002	Switch1	Valence	ICMP
	0.003	Switch1	Valence	ICMP
	0.003	Valence	FAI	ICMP

Cette capture montre que le paquet ICMP envoyé par PC0 (192.168.1.3) à destination de PC4 (192.168.2.3) est encapsulé pour traverser le tunnel IPsec. On peut voir que la source et la destination IP au niveau de la couche 3 sont modifiées pour refléter les adresses des routeurs (200.1.2.1 vers 200.1.1.1). Cette encapsulation confirme que le paquet est protégé par le

tunnel VPN. L'indication "ESP encrypts the received packet" prouve que le chiffrement est bien appliqué sur ce paquet.

The screenshot shows the 'PDU Information at Device: Valence' window. It has three tabs: 'OSI Model', 'Inbound PDU Details', and 'Outbound PDU Details'. The 'Inbound PDU Details' tab is active. It displays the following information:

- At Device: Valence
- Source: PC0
- Destination: PC4

In Layers:

- Layer 7
- Layer 6
- Layer 5
- Layer 4
- Layer 3: IP Header Src. IP: 192.168.1.3, Dest. IP: 192.168.2.3 ICMP Message Type: 8
- Layer 2: Ethernet II Header 0007.EC48.9E64 >> 0000.8A64.0D01
- Layer 1: Port FastEthernet0/0

Out Layers:

- Layer 7
- Layer 6
- Layer 5
- Layer 4
- Layer 3: IP Header Src. IP: 200.1.2.1, Dest. IP: 200.1.1.1
- Layer 2: HDLC Frame HDLC
- Layer 1: Port(s): Serial1/0

Below the layers, a list of 11 steps describes the packet's journey:

1. The routing table finds a routing entry to the destination IP address.
2. The destination network can be reached via 200.1.2.254.
3. The device decrements the TTL on the packet.
4. The traffic is interesting traffic and needs to be encrypted and encapsulated in IPsec PDUs.
5. The packet is getting encrypted and encapsulated in IPsec PDUs.
6. ESP encrypts the received packet.
7. The device encapsulates the data into an IP packet.
8. The device looks up the destination IP address in the routing table.
9. The routing table finds a routing entry to the destination IP address.
10. The destination network can be reached via 200.1.2.254.
11. An IPSEC (ESP/AH) message is sending out of Serial1/0.

At the bottom, there are buttons: 'Challenge Me', '<< Previous Layer', and 'Next Layer >>'.

Dans cette capture, le paquet traverse le routeur FAI, où il conserve ses informations encapsulées (adresse source 200.1.2.1 et destination 200.1.1.1). Cela indique que le paquet continue de passer par le tunnel IPsec sans altération. L'indication "An IPSEC (ESP/AH) message is sending out of Serial0/0/0" montre que le paquet est encore sécurisé en transit.

Ces vérifications permettent de s'assurer que le trafic entre les deux réseaux est correctement encapsulé et sécurisé par IPsec lors de son passage par les différents équipements du réseau.

The screenshot shows the 'PDU Information at Device: FAI' window. It has three tabs: 'OSI Model', 'Inbound PDU Details', and 'Outbound PDU Details'. The 'Inbound PDU Details' tab is active. It displays the following information:

- At Device: FAI
- Source: PC0
- Destination: PC4

In Layers:

- Layer 7
- Layer 6
- Layer 5
- Layer 4
- Layer 3: IP Header Src. IP: 200.1.2.1, Dest. IP: 200.1.1.1
- Layer 2: HDLC Frame HDLC
- Layer 1: Port Serial0/0/1

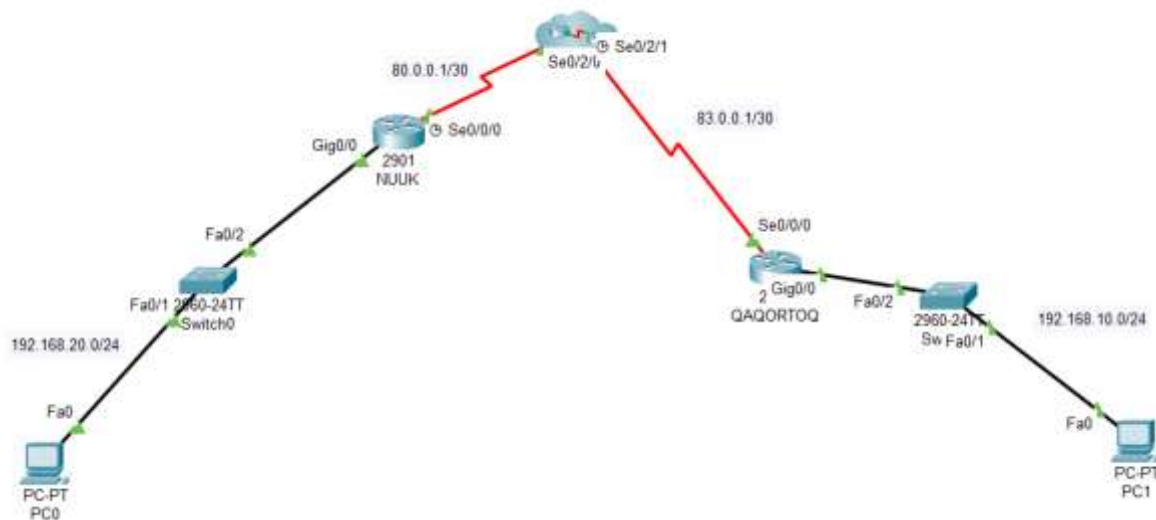
Out Layers:

- Layer 7
- Layer 6
- Layer 5
- Layer 4
- Layer 3: IP Header Src. IP: 200.1.2.1, Dest. IP: 200.1.1.1
- Layer 2: HDLC Frame HDLC
- Layer 1: Port(s): Serial0/0/0

Below the layers, a list of 4 steps describes the packet's journey:

1. The routing table finds a routing entry to the destination IP address.
2. The destination network is directly connected. The device sets destination as the next-hop.
3. The device decrements the TTL on the packet.
4. An IPSEC (ESP/AH) message is sending out of Serial0/0/0.

4. VPN SITE A SITE, APPLICATION



j'ai suivi la première étape qui consiste à activer le module securityk9 sur le routeur. Pour cela, je suis entré en mode de configuration globale avec la commande `conf t`, puis j'ai activé le module avec la commande suivante :

```
license boot module c2900 technology-package securityk9.
```

```
NUUK
Physical Config CLI Attributes
IOS Command Line Interface

NUUK#conf t
Enter configuration commands, one per line. End with CNTL/Z.
NUUK(config)#license boot module c2900 technology-package securityk9
PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR
LICENSE KEY PROVIDED FOR ANY CISCO PRODUCT FEATURE OR USING SUCH
PRODUCT FEATURE CONSTITUTES YOUR FULL ACCEPTANCE OF THE FOLLOWING
TERMS. YOU MUST NOT PROCEED FURTHER IF YOU ARE NOT WILLING TO BE BOUND
BY ALL THE TERMS SET FORTH HEREIN.

Use of this product feature requires an additional license from Cisco,
together with an additional payment. You may use this product feature
on an evaluation basis, without payment to Cisco, for 60 days. Your use
of the product, including during the 60 day evaluation period, is
subject to the Cisco end user license agreement
http://www.cisco.com/en/US/docs/general/warranty/English/EUWEN_.html
If you use the product feature beyond the 60 day evaluation period, you
must submit the appropriate payment to Cisco for the license. After the
60 day evaluation period, your use of the product feature will be
governed solely by the Cisco end user license agreement (link above),
together with any supplements relating to such product feature. The
above applies even if the evaluation license is not automatically
terminated and you do not receive any notice of the expiration of the
evaluation period. It is your responsibility to determine when the
evaluation period is complete and you are required to make payment to
Cisco for your use of the product feature beyond the evaluation period.

Your acceptance of this agreement for the software features on one
product shall be deemed your acceptance with respect to all such
software on all Cisco products you purchase which includes the same
software. (The foregoing notwithstanding, you must purchase a license
for each software feature you use past the 60 days evaluation period,
so that if you enable a software feature on 1000 devices, you must
purchase 1000 licenses for use past the 60 day evaluation period.)

Activation of the software command line interface will be evidence of
your acceptance of this agreement.

ACCEPT? [yes/no]: yes
% use 'write' command to make license boot config take effect on next boot
NUUK(config)#
```

Le message affiché précise que ce module est activé en mode évaluation pour une durée de 60 jours. J'ai accepté les termes en tapant `yes`. Cela permet d'activer temporairement les fonctionnalités avancées nécessaires pour configurer un VPN. À la fin, le message me rappelle

de sauvegarder la configuration et de redémarrer pour appliquer les changements, ce que je fais dans les étapes suivantes.

`copy run start` pour sauvegarder la configuration courante dans la mémoire de démarrage. Cette étape est cruciale pour éviter de perdre les modifications en cas de redémarrage inattendu. Ensuite, j'ai redémarré le routeur avec la commande `reload`.

```
NUUK#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
NUUK#reload
Proceed with reload? [confirm]
System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by Cisco Systems, Inc.
Total memory size = 512 MB - On-board = 512 MB, DIMM0 = 0 MB
CISCO2901/K9 platform with 524288 Kbytes of main memory
Main memory is configured to 72/-1(On-board/DIMM0) bit mode with ECC disabled

Readonly ROMMON initialized

program load complete, entry point: 0x80803000, size: 0x1b340
program load complete, entry point: 0x80803000, size: 0x1b340

IOS Image Load Test

Digitally Signed Release Software
program load complete, entry point: 0x81000000, size: 0x3bcd3d8
Self decompressing the image :
#####
```

Lors du redémarrage, j'ai **confirmé avec** `confirm` pour que le processus s'exécute. Le routeur a initialisé le système et a chargé l'image IOS. C'est cette opération qui active réellement le module `securityk9` que j'avais configuré. Cette étape est indispensable pour valider les changements apportés.

```
NUUK#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.1(4)M4, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team

ROM: System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
cisco2901 uptime is 1 minutes, 20 seconds
System returned to ROM by power-on
System image file is "flash0:c2900-universalk9-mz.SPA.151-1.M4.bin"
Last reload type: Normal Reload

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wvl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.
Cisco CISC02901/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
2 Gigabit Ethernet interfaces
2 Low-speed serial(sync/async) network interface(s)
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)
```

Après le redémarrage, j'ai utilisé la commande `show version` pour vérifier que le module `securityk9` était bien activé. Cette commande me permet de confirmer que l'image IOS `c2900-`

universalk9-mz.SPA.151-4.M4.bin est bien en cours d'utilisation, ce qui inclut les fonctionnalités nécessaires pour configurer un VPN.

```
License Info:

License UDI:

-----
Device#    PID                      SN
-----
*0         CISCO2901/K9                 FTX152415L5

Technology Package License Information for Module:'c2900'

-----
Technology    Technology-package    Technology-package
Current      Type                Next reboot
-----
ipbase        ipbasek9             Permanent          ipbasek9
security      securityk9            Evaluation          securityk9
uc            disable              None               None
data          disable              None               None

Configuration register is 0x2102
```

la commande show license pour afficher les détails des licences installées sur le routeur. Cette capture confirme que le module securityk9 est actif en mode évaluation, comme indiqué sous "Technology-package Current: Evaluation".

La licence affichée sur le routeur, **securityk9**, est en mode évaluation, comme indiqué sous "Type: Evaluation". Cela signifie qu'elle est active temporairement pour une durée de 60 jours. Elle permet d'utiliser des fonctionnalités de sécurité avancées, telles que la configuration de VPN (IPsec, SSL), le chiffrement, et les pare-feux. Après cette période, une licence permanente devra être acquise pour continuer à bénéficier de ces services.

I) ACTIVEZ ISAKMP (IKE - INTERNET KEY EXCHANGE) :

```
NUUK>en
NUUK#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
NUUK(config)#crypto isakmp policy 1
NUUK(config-isakmp)# encr aes
NUUK(config-isakmp)# authentication pre-share
NUUK(config-isakmp)# group 5
NUUK(config-isakmp)#
```

m) DEFINITION DE LA CLE D'ÉCHANGE VPNKEY

Une clé d'échange VPNKEY est un composant essentiel dans un réseau privé virtuel (VPN). Elle sert à établir un canal sécurisé entre deux entités en permettant le chiffrement et le déchiffrement des données. Cette clé peut être utilisée dans différents protocoles pour garantir la confidentialité, l'intégrité, et l'authentification des communications.

Fonctionnement des échanges de clés

1. Clés symétriques

- **Principe** : Une seule clé est utilisée pour chiffrer et déchiffrer les données. Cette clé doit être partagée entre les deux parties de manière sécurisée avant le début de la communication.
- **Exemple de protocole** : AES (Advanced Encryption Standard).
- **Processus** :

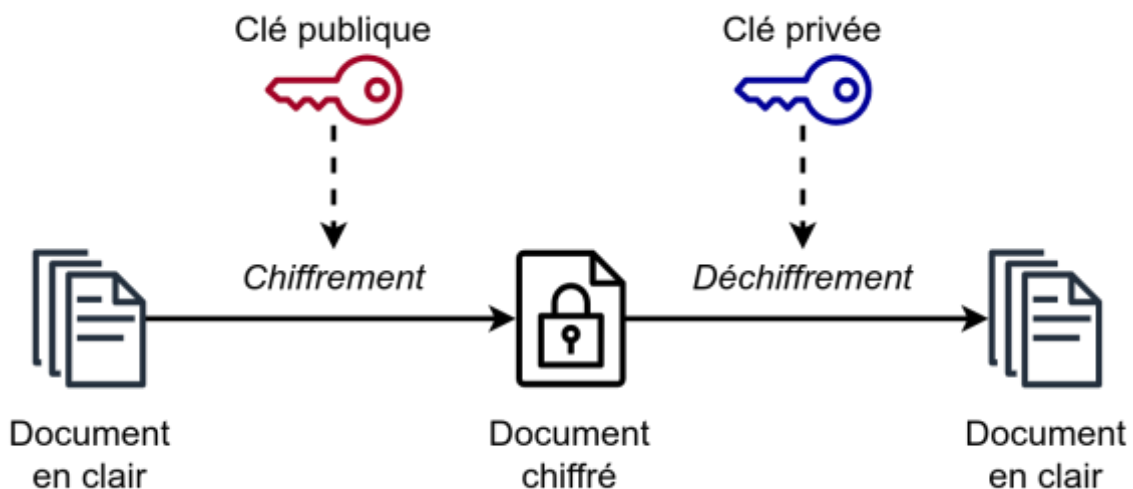
Une clé symétrique est générée par l'une des parties (par exemple, le serveur).

Cette clé est transmise à l'autre partie de manière sécurisée (par exemple, via un échange de clé asymétrique).

Les deux parties utilisent cette clé pour chiffrer et déchiffrer les données.

2. Clés asymétriques

- **Principe** : Deux clés différentes sont utilisées : une clé publique (connue de tous) et une clé privée (secrète). Ce système repose sur des algorithmes comme RSA ou ECC.
- **Exemple de protocole** : TLS (Transport Layer Security).



Pour configurer les clés d'échange dans un VPN avec cette commande, voici les étapes et explications :

```
crypto isakmp key VPNKEY address 83.0.0.1
NUUK(config-isakmp)#crypto isakmp key VPNKEY address 83.0.0.1
```

n) CONFIGURATION DES METHODES DE CRYPTAGE ET D'AUTHENTIFICATION :

Pour configurer un **transform-set** pour un tunnel VPN IPsec avec la commande affichée, voici les explications et les étapes associées :

```
crypto ipsec transform-set CRYPTSET esp-aes esp-sha-hmac
NUUK(config)#crypto ipsec transform-set CRYPTSET esp-aes esp-sha-hmac
```

o) MODIFICATION DES ACL POUR LE NAT :

Passons à la configuration des ACL, nous éditerons dans un premier temps l'ACL du NAT déjà exist

Création de l'ACL étendue nommée :

```
ip access-list extended VPN
permit ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
```

ip access-list extended VPN : Définit une liste de contrôle d'accès (ACL) nommée "VPN".

permit ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255 : Permet le trafic IP entre le réseau source **192.168.20.0/24** et le réseau destination **192.168.10.0/24**.

ACL numérotée avec remarques et règles supplémentaires :

```
access-list 100 remark ==[Services VPN]==-
access-list 100 deny ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
access-list 100 permit ip 192.168.20.0 0.0.0.255 any
```

Application de l'ACL pour la NAT :

```
ip nat inside source list 100 interface Serial0/0/0 overload
```

Associe l'ACL **100** à la NAT dynamique avec surcharge (PAT).

Cette règle NAT permet au réseau défini dans l'ACL 100 de partager une seule adresse IP publique sur l'interface Serial0/0/0.

```
Enter configuration commands, one per line. End with CTRL-Z.
NUUK(config)#ip access-list extended VPN
NUUK(config-ext-nacl)# permit ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
NUUK(config-ext-nacl)#access-list 100 remark ==[Services VPN]==-
NUUK(config)#access-list 100 deny ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
NUUK(config)#access-list 100 permit ip 192.168.20.0 0.0.0.255 any
NUUK(config)#ip nat inside source list 100 interface Serial0/0/0 overload
NUUK(config)#
```

p) PROBLEME DES ACL SUR PACKET TRACER :

Les ACL ne se mettent pas à jour automatiquement.

Sur **Packet Tracer**, lorsqu'une ACL est modifiée, la nouvelle configuration ne s'applique pas immédiatement si l'ACL existante est déjà en cours d'utilisation.

Raison : Packet Tracer ne supporte pas la mise à jour dynamique des ACL utilisées par des processus comme la NAT.

q) SOLUTION : REMPLACER L'ACL POUR FORCER SA MISE A JOUR

Pour appliquer les nouvelles modifications, il faut retirer temporairement l'ACL de la configuration en suivant ces étapes :

Désassocier l'ACL de la NAT :

```
no ip nat inside source list 100 interface Serial0/0/0 overload
```

Reconfigurer ou mettre à jour l'ACL :

```
no access-list 100
```

Recréez l'ACL avec les nouvelles règles :

```
access-list 100 remark ==[Services VPN]==-
access-list 100 deny ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
access-list 100 permit ip 192.168.20.0 0.0.0.255 any
```

Rattacher l'ACL mise à jour à la NAT :

```
ip nat inside source list 100 interface Serial0/0/0 overload
NUUK(config)#no ip nat inside source list 100 interface Serial0/0/0 overload
NUUK(config)#no access-list 100
NUUK(config)#access-list 100 remark ==[Services VPN]==-
NUUK(config)#access-list 100 deny ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
NUUK(config)#access-list 100 permit ip 192.168.20.0 0.0.0.255 any
NUUK(config)#
NUUK(config)#ip nat inside source list 100 interface Serial0/0/0 overload
NUUK(config)#
```

Nous nous attaquons à présent à rassembler les différents éléments créés ci-dessus pour en faire une cryptomap que l'on positionnera sur l'interface outside serial 0/0/0. Les commandes à utiliser pourraient ressembler aux suivantes :

```
crypto map VPNMAP 10 ipsec-isakmp
NUUK(config)#crypto map VPNMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
NUUK(config-crypto-map)#
```

```
set peer 83.0.0.1
set transform-set CRYPTSET
match address VPN
exit
interface serial 0/0/0
crypto map VPNMAP
NUUK(config-crypto-map)#set peer 83.0.0.1
NUUK(config-crypto-map)#set transform-set CRYPTSET
NUUK(config-crypto-map)#match address VPN
NUUK(config-crypto-map)#exit
NUUK(config)#interface serial 0/0/0
NUUK(config-if)#crypto map VPNMAP
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
NUUK(config-if)#
```

Commandes appliquées :

```
crypto map CMAP 10 ipsec-isakmp
set peer 83.0.0.1
set transform-set CRYPTSET
match address VPN
interface Serial0/0/0
crypto map CMAP
NUUK(config-if)#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.
NUUK(config-crypto-map)# set peer 83.0.0.1
NUUK(config-crypto-map)# set transform-set CRYPTSET
NUUK(config-crypto-map)# match address VPN
NUUK(config-crypto-map)#interface Serial0/0/0
NUUK(config-if)# crypto map CMAP
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
NUUK(config-if)#
```

5. CONFIGURATION DU ROUTEUR DE QAQORTOQ

Configuration d'ISAKMP :

```
crypto isakmp policy 1
encr aes
authentication pre-share
group 5
QAQORTOQ(config)#crypto isakmp policy 1
QAQORTOQ(config-isakmp)# encr aes
QAQORTOQ(config-isakmp)# authentication pre-share
QAQORTOQ(config-isakmp)# group 5
```

Configuration de la clé d'échange entre les 2 hôtes

```
crypto isakmp key VPNKEY address 80.0.0.1
QAQORTOQ(config)#crypto isakmp key VPNKEY address 80.0.0.1
```

Configuration du cryptage et de la méthode d'authentification

```
crypto ipsec transform-set CRYPTSET esp-aes esp-sha-hmac
QAQORTOQ(config)#crypto ipsec transform-set CRYPTSET esp-aes esp-sha-hmac
```

r) CONFIGURATION DE L'ACL LAN (CELLE DU NAT)

Création de l'ACL nommé VPN

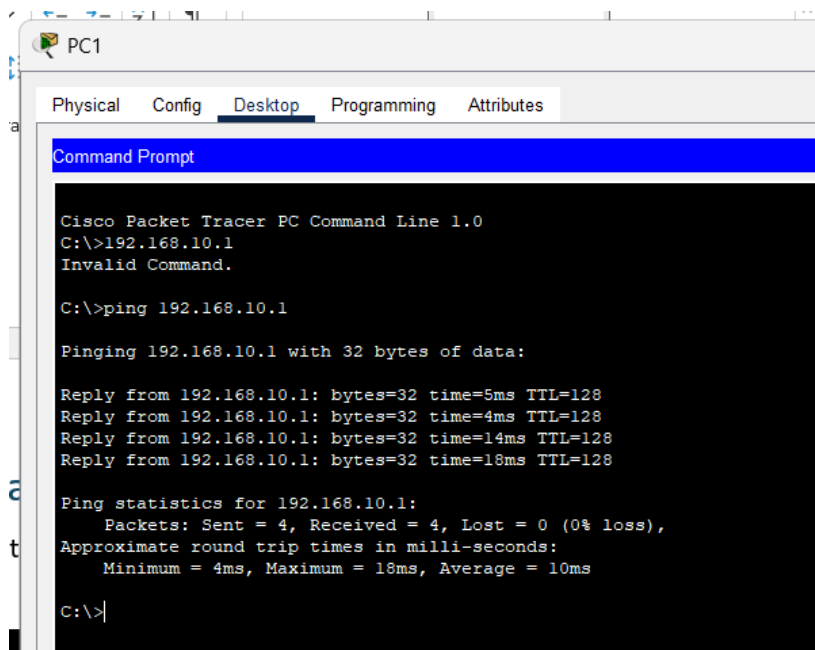
```
ip access-list extended VPN
 permit ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
access-list 100 remark -=[Services VPN]=-
access-list 100 deny ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
access-list 100 permit ip 192.168.10.0 0.0.0.255 any
ip nat inside source list 100 interface Serial0/0/0 overload
QAQORTOQ(config)#ip access-list extended VPN
QAQORTOQ(config-ext-nacl)# permit ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
QAQORTOQ(config-ext-nacl)#access-list 100 remark -=[Services VPN]=-
QAQORTOQ(config)#access-list 100 deny ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
QAQORTOQ(config)#access-list 100 permit ip 192.168.10.0 0.0.0.255 any
QAQORTOQ(config)#ip nat inside source list 100 interface Serial0/0/0 overload
```

Création de la Cryptomap

```
crypto map CMAP 10 ipsec-isakmp
 set peer 80.0.0.1
 set transform-set CRYPTSET
 match address VPN
interface Serial0/0/0
 crypto map CMAP
QAQORTOQ(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.
QAQORTOQ(config-crypto-map)# set peer 80.0.0.1
QAQORTOQ(config-crypto-map)# set transform-set CRYPTSET
QAQORTOQ(config-crypto-map)# match address VPN
QAQORTOQ(config-crypto-map)#interface Serial0/0/0
QAQORTOQ(config-if)# crypto map CMAP
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

s) TESTS ET VERIFICATIONS

Bien, nous avons maintenant terminé la configuration. Nous allons passer à la vérification en utilisant des commandes *ping* plutôt que des tests avec des paquets *tracert*. En effet, le simulateur ne prend pas en charge le scénario impliquant l'IPsec, ce qui rend les *ping* plus adaptés pour cette validation.



```
PC1
Physical Config Desktop Programming Attributes
Command Prompt

Cisco Packet Tracer PC Command Line 1.0
C:\>192.168.10.1
Invalid Command.

C:\>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time=5ms TTL=128
Reply from 192.168.10.1: bytes=32 time=4ms TTL=128
Reply from 192.168.10.1: bytes=32 time=14ms TTL=128
Reply from 192.168.10.1: bytes=32 time=18ms TTL=128

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 18ms, Average = 10ms

C:\>|
```



```
QAQORTQ#sh crypto ipsec sa

interface: Serial0/0/0
  Crypto map tag: CMAP, local addr 83.0.0.1

  protected vrf: (none)
  local ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (192.168.20.0/255.255.255.0/0/0)
  current_peer 80.0.0.1 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.: 83.0.0.1, remote crypto endpt.:80.0.0.1
  path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
  current outbound spi: 0x0(0)

  inbound esp sas:

  inbound ah sas:

  inbound pcg sas:

  outbound esp sas:
```

La commande `sh crypto ipsec sa` montre l'état de l'association de sécurité IPsec sur l'interface *Serial0/0/0*. L'adresse locale est **83.0.0.1**, et le pair distant est **80.0.0.1** (port 500). Les identifiants locaux et distants sont respectivement **192.168.10.0/24** et **192.168.20.0/24**. Aucun paquet n'a été encapsulé ou décapsulé, et les SPI sont à zéro, indiquant qu'aucune session active n'est encore en cours. Les paramètres MTU sont définis à 1500, mais aucun trafic chiffré ou authentifié n'est visible.

6. PROPOSITION DE SOLUTION EN CAS DE DYSFONCTIONNEMENT :

La commande `show crypto isakmp sa` permet de vérifier l'état des associations de sécurité ISAKMP. Ici, aucune session active n'est affichée, mais cela ne signifie pas nécessairement un problème si aucune négociation n'a encore été initiée.

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

IPv6 Crypto ISAKMP SA
```

La commande `show ip route` confirme que les routes sont correctement configurées. Les réseaux **83.0.0.0/30** et **192.168.10.0/24** sont bien directement connectés, respectivement à l'interface *Serial0/0/0* et à l'interface *GigabitEthernet0/0*. Une route par défaut (S*) est également configurée et pointe vers l'interface *Serial0/0/0*. Ces routes valident la connectivité du réseau et confirment que le routage est correctement établi.

```
QAQORTQ#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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, Is - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O    33.0.0.0/30 is directly connected, Serial0/0/0
L    33.0.0.1/32 is directly connected, Serial0/0/0
L    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.10.0/24 is directly connected, GigabitEthernet0/0
L    192.168.10.254/32 is directly connected, GigabitEthernet0/0
S*   0.0.0.0/0 is directly connected, Serial0/0/0
```

Activation du mode débogage ISAKMP et IPsec

Pour diagnostiquer le problème, le mode débogage a été activé avec les commandes debug crypto isakmp et debug crypto ipsec. Cela permet de surveiller en temps réel les processus liés à ISAKMP et IPsec, notamment la négociation des associations de sécurité et l'échange de clés. Ces commandes sont essentielles pour identifier les erreurs ou dysfonctionnements potentiels dans la configuration ou la communication IPsec.

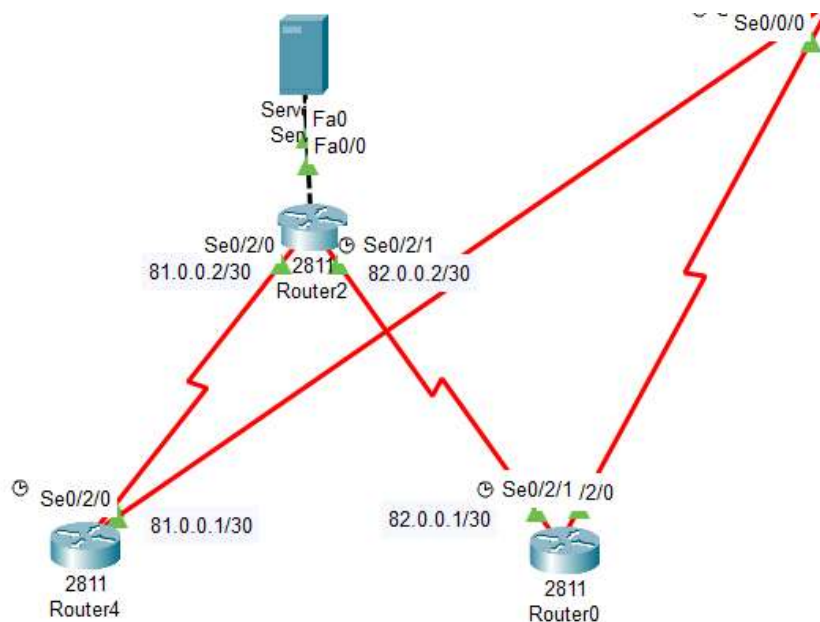
```
QAQORTQ#debug crypto isakmp
Crypto ISAKMP debugging is on
QAQORTQ#debug crypto ipsec
Crypto IPSEC debugging is on
QAQORTQ#
```



Après activation du mode débogage et validation, la commande show crypto ipsec sa montre que la communication fonctionne correctement. On observe que 3 paquets ont été encapsulés, chiffrés et décapsulés, ce qui confirme que les échanges IPsec sont désormais opérationnels. Les ACL et les paramètres IPsec sont bien appliqués, et le trafic est sécurisé comme attendu.

7. ROUTES

La dernière étape consiste à vérifier les routes configurées sur le WAN. Cette vérification permet de s'assurer que le problème lié aux paquets (*tracer* ou autres enveloppes) ne provient pas d'une mauvaise configuration ou absence de route vers la destination via le WAN. Il est important de confirmer que les routes par défaut et spécifiques au WAN sont correctement définies pour garantir une connectivité optimale.



La configuration des routes WAN a révélé une petite erreur dans les adresses réseau :

- **Routeur 2 (Se0/2/0) : 81.0.0.2/30**
- **Routeur 4 (Se0/2/0) : 81.0.0.1/30**
- **Routeur 2 (Se0/2/1) : 82.0.0.2/30**
- **Routeur 0 (Se0/2/1) : 82.0.0.1/30**

Une légère modification a été apportée dans la configuration des routes pour s'assurer que toutes les adresses et masques correspondent correctement. Une fois cette correction effectuée, les tests ICMP (ping) ont confirmé une connectivité réussie entre les routeurs, indiquant que le routage WAN est maintenant opérationnel.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	Router2	Router4	ICMP		0.000	N	0	(edit)	(delete)
	Successful	Router2	Router0	ICMP		0.000	N	1	(edit)	(delete)
	Successful	Router0	Router4	ICMP		0.000	N	2	(edit)	(delete)

8. OPENVPN

Pour voir OPEN VPN :

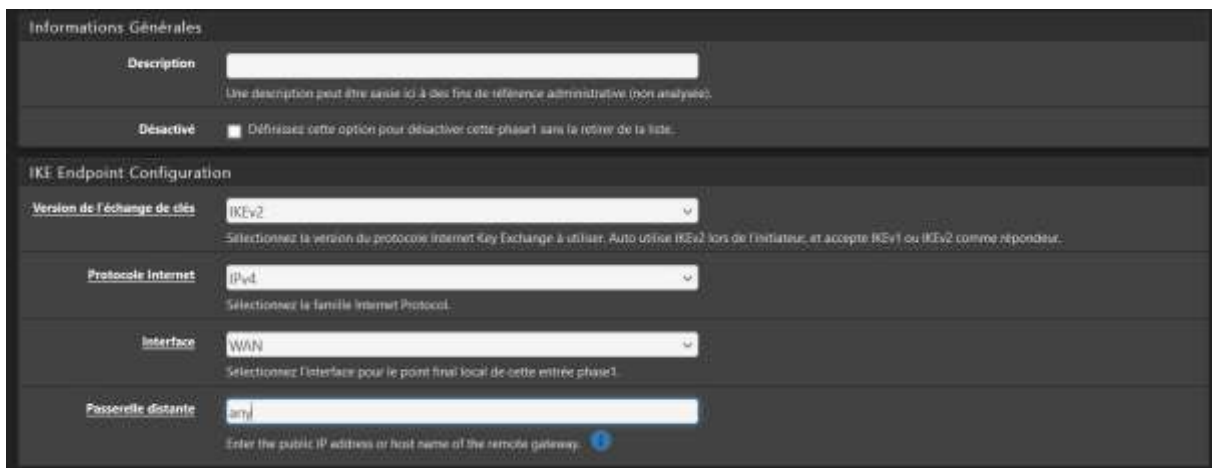
[PfSens_Eloham_caron.docx](#)

9. MISE EN PLACE D'IPSEC

Pour configurer l'IPsec : **VPN > IPsec** dans le menu principal de pfSense. Cette action permet d'accéder à la configuration des tunnels IPsec afin de sécuriser les communications réseau entre différents points.



Je procède à la configuration des informations générales et du point de terminaison IKE. Pour cela, je choisis IKEv2 comme version de l'échange de clés, car elle est plus moderne et sécurisée que IKEv1. Je sélectionne IPv4 comme protocole Internet, ce qui correspond au standard le plus courant pour le trafic réseau. Ensuite, je définis l'interface sur WAN pour que le trafic passe par l'interface externe. Enfin, dans le champ Passerelle distante, j'inscris "any" pour permettre des connexions provenant de n'importe quelle adresse IP distante.

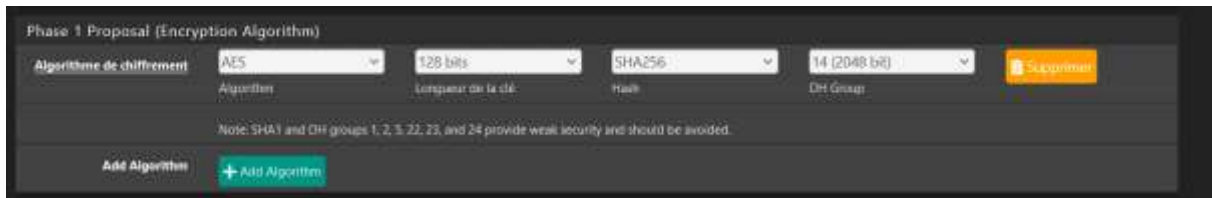


Dans cette section, je configure la proposition de phase 1 pour l'authentification. Je choisis la méthode **PSK Mutuel**, qui repose sur une clé partagée pour sécuriser le tunnel, et doit correspondre à celle configurée sur le pair distant. Pour l'identifiant local, je sélectionne **Mon adresse IP**, tandis que pour l'identifiant du pair, j'indique **Adresse IP distante**. Ensuite, j'entre une **Clé Pré-Partagée** générée automatiquement, qui doit être longue et aléatoire afin de renforcer la sécurité du tunnel. Cette clé doit être identique des deux côtés pour que la connexion soit établie correctement.

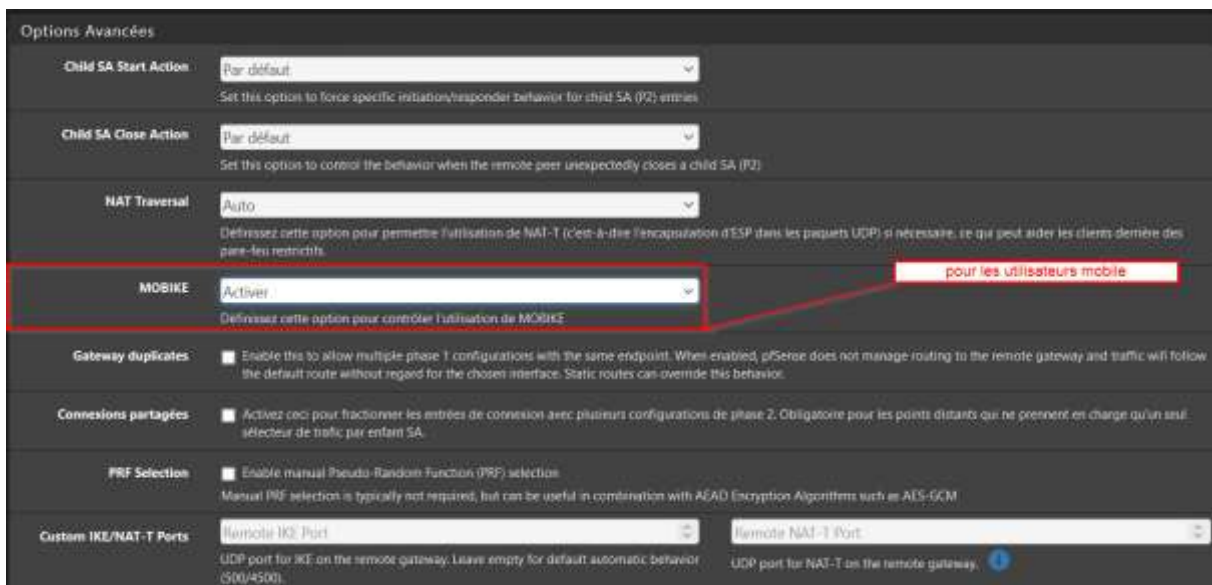


Cette version alternative de la configuration de phase 1 permet de prendre en charge plusieurs utilisateurs grâce à l'authentification **PSK + Xauth Mutuel**. Cette méthode ajoute une couche d'authentification utilisateur après la connexion initiale avec une clé pré-partagée. Je choisis **Mon adresse IP** comme identifiant local, tandis que l'identifiant de pair est configuré sur **Adresse IP distante** pour identifier le serveur VPN. La **Clé Pré-Partagée** est une chaîne longue et aléatoire nécessaire pour sécuriser le tunnel, qui doit être identique des deux côtés. Cette configuration est idéale pour un accès VPN multi-utilisateur tout en garantissant une authentification renforcée par mot de passe pour chaque utilisateur connecté.

Je définis maintenant les paramètres de chiffrement pour la phase 1. Je choisis l'algorithme **AES**, qui est reconnu pour sa robustesse et sa fiabilité. Pour une meilleure performance, j'opte pour une longueur de clé de **128 bits**. Je sélectionne le hash **SHA256** afin d'assurer l'intégrité des données échangées. Pour le groupe Diffie-Hellman, je choisis le groupe **14 (2048 bits)**, garantissant ainsi une sécurité renforcée lors de l'échange des clés.



Enfin, je configure les options avancées pour le tunnel IPsec. J'active l'option **MOBIKE**, ce qui permet aux utilisateurs mobiles de changer d'adresse IP sans interruption du tunnel VPN. Cela est particulièrement utile pour les personnes qui alternent entre différentes connexions réseau, comme le Wi-Fi et la 4G. Grâce à cette configuration, le tunnel reste stable et fonctionnel malgré les changements de réseau.



On voit que nos deux tunnels IPsec ont bien été créés. Le premier tunnel, pour l'**Accès Local**, utilise l'authentification **Mutual PSK** avec un chiffrement **AES 128 bits** et une intégrité assurée par **SHA256**. La passerelle distante est configurée sur "any", ce qui permet des connexions depuis n'importe quelle adresse IP.

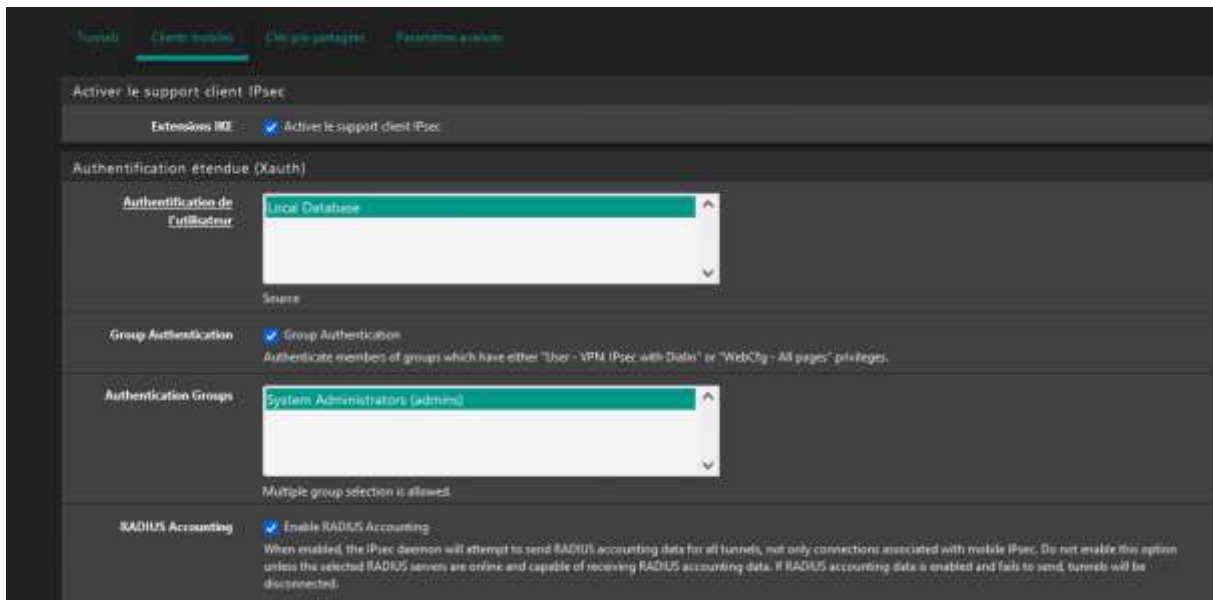
Le deuxième tunnel, destiné à l'**Accès Distant** pour les utilisateurs mobiles, utilise l'authentification **Mutual PSK + Xauth** pour une sécurité renforcée. Il adopte le même chiffrement **AES 128 bits** et l'intégrité **SHA256**. La passerelle est configurée pour accepter les **clients mobiles**.

Ces configurations assurent que nos tunnels IPsec sont prêts à fonctionner pour des besoins de connexion locale et distante.



t) CONFIGURATION DES CLIENTS-MOBILES

J'active le support client IPsec en cochant “**Activer le support client IPsec**”. Cela permet aux utilisateurs mobiles de se connecter au VPN via IPsec. Pour l'authentification étendue (Xauth), je choisis “**Local Database**” comme source d'authentification, afin que les utilisateurs soient authentifiés localement sur le pare-feu. J'active également “**Group Authentication**” et sélectionne le groupe “**System Administrators (admins)**” pour restreindre l'accès aux administrateurs autorisés. Cela garantit une sécurité renforcée en limitant les connexions aux personnes ayant les droits nécessaires.



Ici, je configure les utilisateurs autorisés pour le VPN IPsec. Trois utilisateurs sont déjà créés : **Bruno**, **Eloham**, et **admin**. Ils sont tous activés et appartiennent aux groupes “**Administrateur**” et “**admins**”, ce qui leur donne les privilèges nécessaires pour accéder au VPN. Cette configuration permet de gérer facilement les droits d'accès en ajoutant ou en supprimant des utilisateurs selon les besoins.



Dans cette section, je configure le **pool d'adresses virtuelles** avec le réseau **172.31.1.0/24**. Cela permet d'attribuer automatiquement des adresses IP aux clients mobiles connectés via

le VPN. Je remplis également le champ **Domaine DNS par défaut** avec l'adresse **10.0.104.1** pour que les clients utilisent ce domaine par défaut pour leurs requêtes DNS. Cela facilite l'accès aux ressources internes sans configuration manuelle côté client.

The screenshot shows the 'Configuration client (mode-clg)' interface. It includes several sections with checkboxes and input fields:

- Pool d'adresses virtuelles:** Checked. Input: 172.31.1.0, 24.
- Pool d'adresses IPv6 virtuelles:** Not checked.
- RADIUS IP address priority:** Checked. Note: IPv4/IPv6 address pool is used if address is not supplied by RADIUS server.
- RADIUS Advanced Parameters:** Not checked. Note: Set Advanced RADIUS parameters. May only be required when using 2FA/MFA with RADIUS or under high load.
- Liste de réseaux:** Checked.
- Enregistrer le mot de passe Xauth:** Not checked. Note: Autoriser les clients à sauvegarder les mots de passe Xauth (clients Cisco VPN seulement). REMARQUE: avec les clients iPhone, cela ne fonctionne pas lorsqu'il est déployé via l'utilitaire de configuration iPhone, uniquement par saisie manuelle.
- Domaine DNS par défaut:** Checked. Input: 10.0.104.1. Note: Spécifiez le domaine en tant que domaine par défaut DNS.
- DNS partagé:** Checked. Input: 10.0.104.1. Note: REMARQUE: si elle est laissée vierge, et un domaine par défaut est défini, il sera utilisé pour cette valeur.
- Serveurs DNS:** Checked. Note: NOTE: IPv4-mapped IPv6 addresses (ex: fd00:123:4) are not supported.

Je configure les serveurs DNS pour les clients VPN. Je renseigne trois adresses de serveurs DNS : **10.0.104.1**, **1.1.1.1**, et **1.0.0.1**. Cela garantit que les clients disposent de plusieurs options pour résoudre les noms de domaine, ce qui améliore la redondance et la fiabilité des requêtes DNS. Cette configuration assure une connectivité stable et efficace pour les utilisateurs mobiles du VPN.

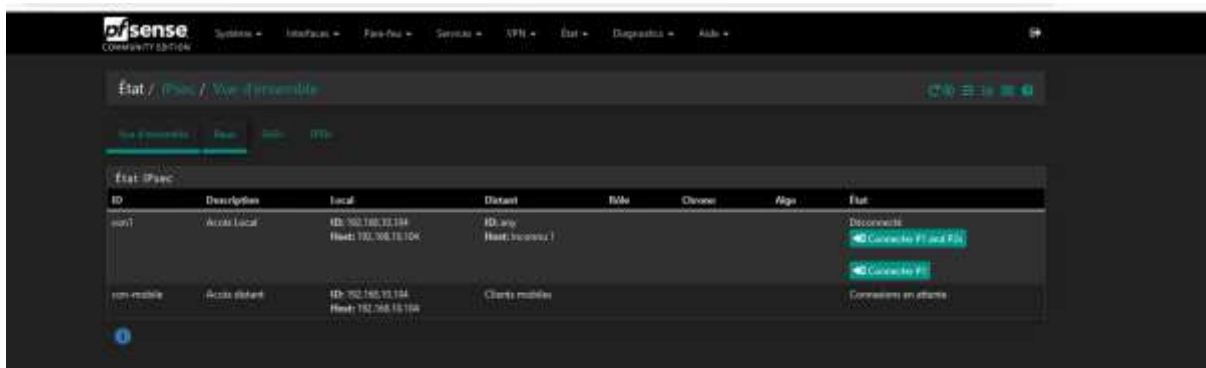
This screenshot shows a detailed view of the DNS configuration section. It includes:

- DNS partagé:** Checked. Input: 10.0.104.1. Note: REMARQUE: si elle est laissée vierge, et un domaine par défaut est défini, il sera utilisé pour cette valeur.
- Serveurs DNS:** Checked. Note: NOTE: IPv4-mapped IPv6 addresses (ex: fd00:123:4) are not supported.
- Server #1:** Input: 10.0.104.1
- Server #2:** Input: 1.1.1.1
- Server #3:** Input: 1.0.0.1
- Server #4:** Empty input field.
- Serveurs WINS:** Not checked. Note: Fournir une liste de serveur WINS aux clients.
- Groupe PFS Phase2:** Not checked. Note: Fournir le groupe PFS Phase2 aux clients (surpasse tous les paramètres de phase2 mobile).
- Bannière de connexion:** Not checked. Note: Renseigner un bannière de connexion pour les clients.

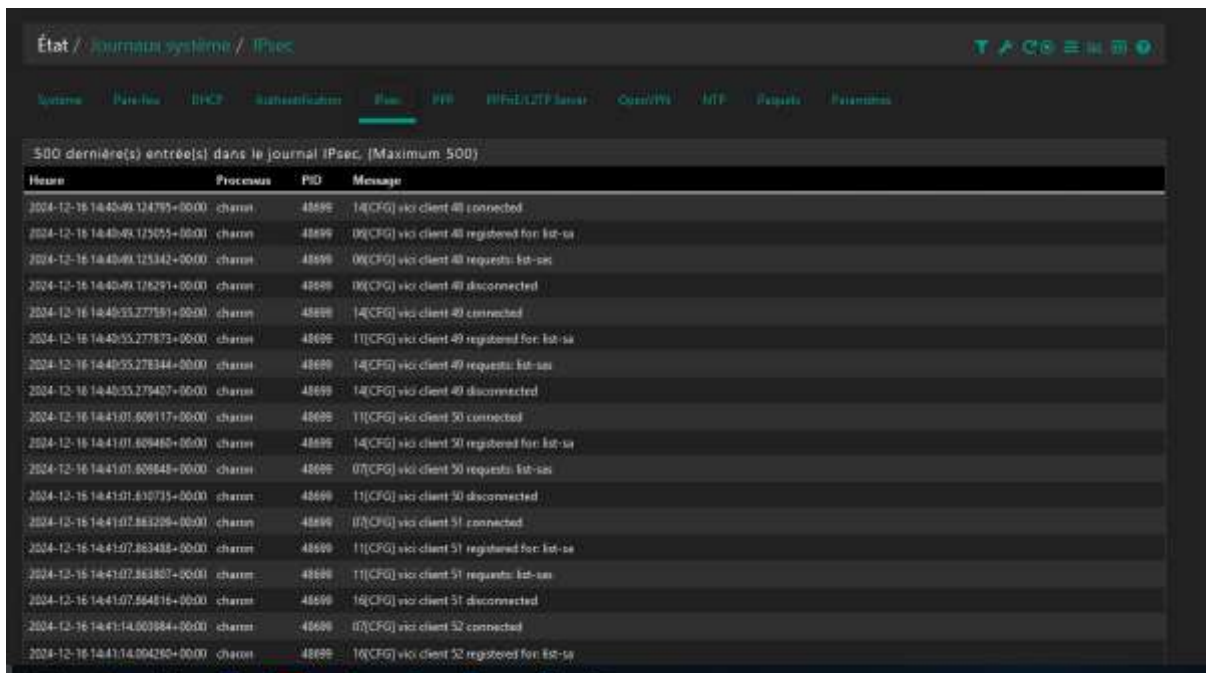
Je configure une **clé pré-partagée** pour l'utilisateur **elohan.caron@gmail.com**. La clé est générée et associée au type **PSK**. Cette clé est nécessaire pour établir des connexions sécurisées avec les clients IPsec. Cela garantit que seuls les utilisateurs disposant de cette clé peuvent s'authentifier et se connecter au VPN.



On voit que les tunnels IPsec sont bien configurés et prêts à fonctionner. Le tunnel **Accès Local** (ID **con1**) est configuré mais actuellement déconnecté. Il est possible de le reconnecter en cliquant sur “**Connecter P1 and P2**”. Le tunnel **Accès Distant** (ID **con-mobile**) destiné aux clients mobiles est en attente de connexions. Cela indique que la configuration des tunnels est en place et prête pour l’établissement de connexions sécurisées.

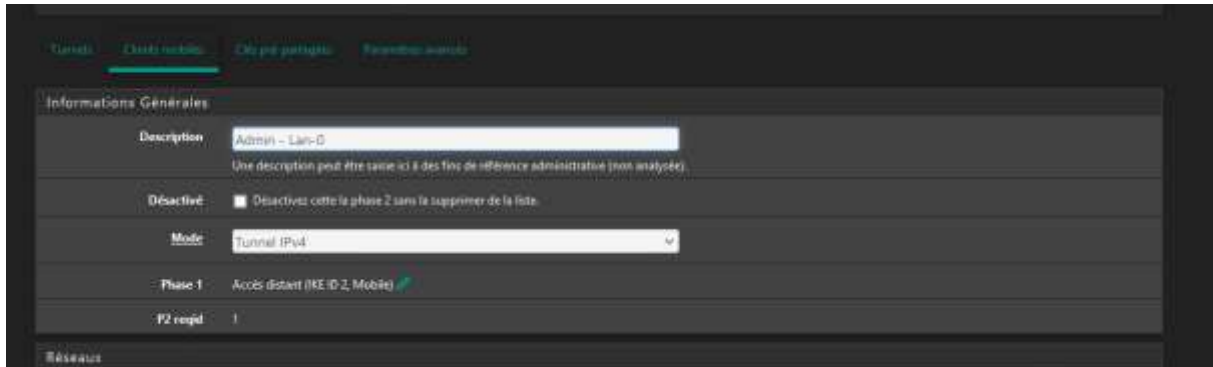


Les journaux de connexion IPsec montrent que des utilisateurs se sont bien connectés au VPN. On voit des messages confirmant que plusieurs clients, comme le **client 48**, **client 49**, et **client 51**, ont établi une connexion avec succès. Les logs indiquent également les enregistrements de ces clients avec les paramètres de sécurité configurés. Cela confirme que le tunnel est opérationnel et que les connexions se déroulent correctement.



u) POOLS IPSEC :

Je configure une connexion IPsec pour les clients mobiles avec une description “**Admin - Lan-0**”. Je choisis le mode **Tunnel IPv4** pour établir un tunnel sécurisé avec les adresses IPv4. La **Phase 1** est définie pour un accès distant (IKE ID 2, Mobile), et le **P2 reqid** est défini sur **1**, ce qui identifie cette configuration pour la phase 2 du tunnel.



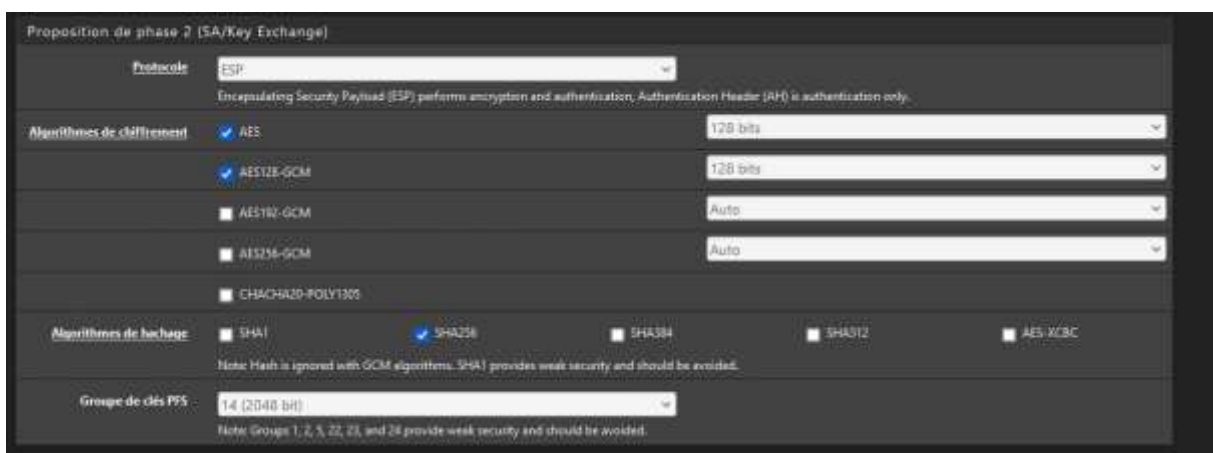
The screenshot shows the 'Informations Générales' tab of the IPsec configuration interface. The 'Description' field is set to 'Admin - Lan-0'. The 'Mode' is set to 'Tunnel IPv4'. The 'Phase 1' is set to 'Accès distant (IKE ID 2, Mobile)'. The 'P2 reqid' is set to '1'.

Je spécifie le **réseau local** pour les clients mobiles, en choisissant **LAN0 subnet** comme réseau accessible pour le tunnel IPsec. Je laisse la **Traduction NAT/BINAT** sur **Aucun**, car aucune traduction d'adresse n'est nécessaire pour ce réseau. Cela permet aux clients de se connecter directement aux ressources internes du sous-réseau LAN0.



The screenshot shows the 'Réseaux' tab of the IPsec configuration interface. The 'Réseau local' is set to 'LAN0 subnet'. The 'Traduction NAT/BINAT' is set to 'Aucun'.

Pour la **phase 2 de l'échange de clés (SA)**, je choisis le protocole **ESP** pour chiffrer et authentifier les données. J'active le chiffrement **AES** avec une longueur de clé de **128 bits** et **AES128-GCM** pour une sécurité renforcée. Comme algorithme de hachage, je sélectionne **SHA256** pour assurer l'intégrité des paquets. Enfin, j'utilise le groupe de clés Diffie-Hellman **14 (2048 bits)** pour sécuriser l'échange de clés.



The screenshot shows the 'Proposition de phase 2 (SA/Key Exchange)' tab of the IPsec configuration interface. The 'Protocole' is set to 'ESP'. The 'Algorithmes de chiffrement' are set to 'AES' (128 bits) and 'AES128-GCM' (128 bits). The 'Algorithmes de hachage' are set to 'SHA256'. The 'Groupe de clés PFS' is set to '14 (2048 bits)'.

Je configure les paramètres d'expiration pour la **phase 2**. La **Life Time** est définie sur **3600 secondes**, ce qui correspond à une heure avant l'expiration du SA enfant. Le **Rekey Time** est réglé sur **3240 secondes** pour renouveler les clés avant l'expiration du SA. Enfin, le **Rand**

Time est fixé à **360 secondes** pour introduire une variation aléatoire et éviter une renégociation simultanée des clés.

Expiration and Replacement

Life Time: 3600
Hard Child SA life time, in seconds, after which the Child SA will be expired. Must be larger than Rekey Time. Cannot be set to the same value as Rekey Time. If left empty, defaults to 115% of Rekey Time. If both Life Time and Rekey Time are empty, defaults to 3600.

Rekey Time: 3600
Time, in seconds, before a Child SA establishes new keys. This works without interruption. Cannot be set to the same value as Life Time. Leave blank to use a default value of 90% Life Time. If both Life Time and Rekey Time are empty, defaults to 3600. Enter a value of 0 to disable, but be aware that when rekey is disabled, connections can be interrupted while new Child SA entries are negotiated.

Rand Time: 300
A random value up to this amount will be subtracted from Rekey Time to avoid simultaneous renegotiation. If left empty, defaults to 10% of Life Time. Enter 0 to disable randomness, but be aware that simultaneous renegotiation can lead to duplicate security associations.

Enregistrer

On voit les différentes configurations des pools pour IPsec. Trois entrées sont présentes :

- **LAN0** avec le protocole **ESP**, le chiffrement **AES128-GCM (128 bits)** et SHA256 pour l'authentification.
- **LAN1** avec **AES128-GCM (128 bits)** et SHA384.
- **LAN2** pour les utilisateurs généraux avec **AES128-GCM (128 bits)** et SHA512.

Ces configurations garantissent des connexions sécurisées pour différents sous-réseaux internes, adaptées à chaque groupe d'utilisateurs et niveau de sécurité requis.

	ID	Mode	Sous-réseau local	Sous-réseau distant	Protocole P2	Transformations P2	Méthodes d'authentification P2	Description	Actions P2
<input type="checkbox"/>	1	tunnel	LAN0		ESP	AES (128 bits), AES128-GCM (128 bits)	SHA256	Admin - Lan-0	
<input type="checkbox"/>	2	tunnel	LAN1		ESP	AES (128 bits), AES128-GCM (128 bits)	SHA384	Admin - Lan-1	
<input type="checkbox"/>	3	tunnel	LAN2		ESP	AES (128 bits), AES128-GCM (128 bits)	SHA512	Utilisateurs	
+ Ajouter P2									

[+ Ajouter P1](#) [Supprimer les P1](#)

v) CONFIGURATION DES REGLES PARE-FEU POUR IPSEC

Je configure ici une règle pare-feu pour IPsec. Dans la section **Source**, je définis une plage de ports allant de **ISAKMP (500)** à **IPsec NAT-T (4500)**, qui sont les ports nécessaires pour l'établissement des connexions IPsec, notamment pour l'échange de clés et le mode NAT-T. Je fais de même pour la section **Destination** afin de m'assurer que le trafic IPsec est autorisé dans les deux sens. Cela permet aux connexions IPsec d'être initiées et maintenues sans blocage.

Source

Source: ☐ Invert match Source Address:

[Masquer les paramètres avancés](#)

La plage de ports source d'une connexion est généralement aléatoire et presque jamais égale au port de destination. Dans la plupart des cas, ce paramètre doit rester à sa valeur par défaut, any.

Plage de port source: À

Spécifiez le port source ou la plage de port pour cette règle. Le champ « A » peut rester vide seulement si le filtre est sur un seul port.

Destination

Destination: ☐ Invert match Destination Address:

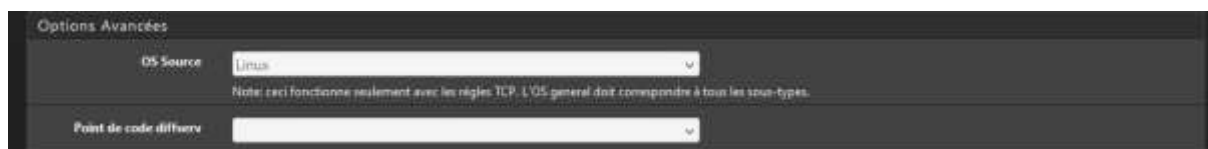
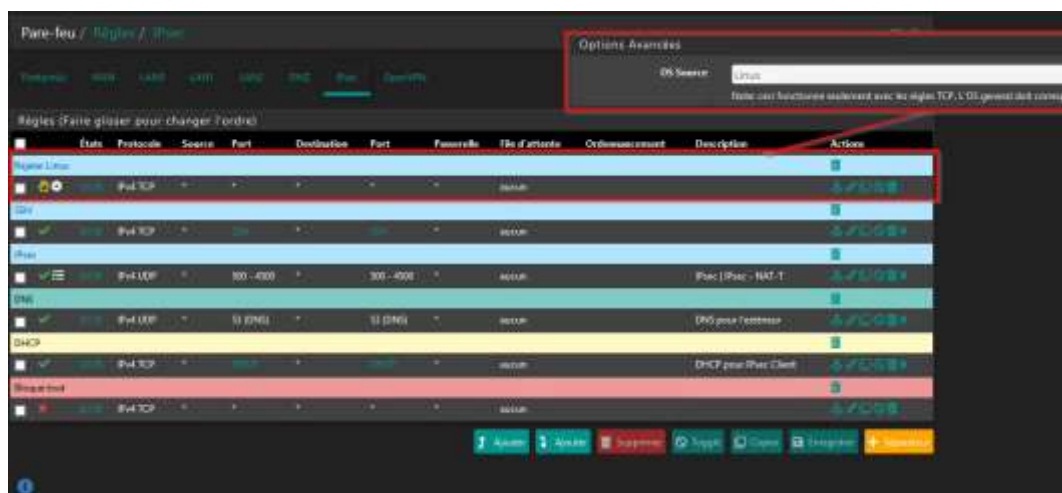
Plage de port de destination: À

Spécifiez le port destination ou la plage de port pour cette règle. Le champ « A » peut rester vide seulement si le filtre est sur un seul port.

J'active l'option de **journalisation** pour cette règle en cochant "**Journaliser les paquets gérés par cette règle**". Cela permet de garder une trace des connexions IPsec autorisées et bloquées, facilitant ainsi le diagnostic et le suivi des activités réseau. Cette journalisation est utile pour surveiller le bon fonctionnement des tunnels et identifier rapidement des problèmes éventuels.



On peut voir l'ensemble des règles pare-feu appliquées à l'interface IPsec. Une règle est configurée pour autoriser le trafic IPsec avec le protocole **UDP** sur les ports **500 à 4500**. Cela garantit que les échanges de clés et le trafic NAT-T passent correctement. Il y a aussi des règles pour le **DNS**, **SSH**, et le **DHCP** pour le client IPsec, assurant une gestion complète des différents besoins réseau. Enfin, une règle de **blocage global** empêche tout trafic non autorisé, renforçant ainsi la sécurité globale du réseau.



Rejet des Postes Linux pour Renforcer la Sécurité

Dans cette configuration, je définis une règle spécifique pour **rejeter le trafic provenant des postes Linux**. Pour cela, je sélectionne "**Linux**" comme **OS Source** dans les **Options Avancées**. Cette règle fonctionne uniquement pour le trafic TCP et permet de bloquer tout appareil utilisant Linux comme système d'exploitation. Cela vise à **réduire les risques d'attaques**, car les systèmes Linux sont couramment utilisés pour des tentatives d'intrusion et des attaques automatisées. En bloquant ces sources, je renforce considérablement la sécurité du réseau en limitant les menaces potentielles.

2024-12-16 16:30:46.260189+00:00	charon	48699	10[CFG] id = 192.168.10.104
2024-12-16 16:30:46.260248+00:00	charon	48699	10[CFG] remote:

On peut voir que les connexions depuis l'extérieur ont bien lieu grâce aux informations affichées dans les logs. Le premier message indique l'ID de l'utilisateur avec

l'adresse **192.168.10.104**, ce qui montre que le client a correctement initié une connexion au VPN. La ligne suivante confirme que le processus **charon** a traité cette requête pour une connexion distante. Cela valide que les clients externes peuvent se connecter avec succès au réseau via le tunnel IPsec configuré.

w) CONCLUSION SUR L'IPSEC

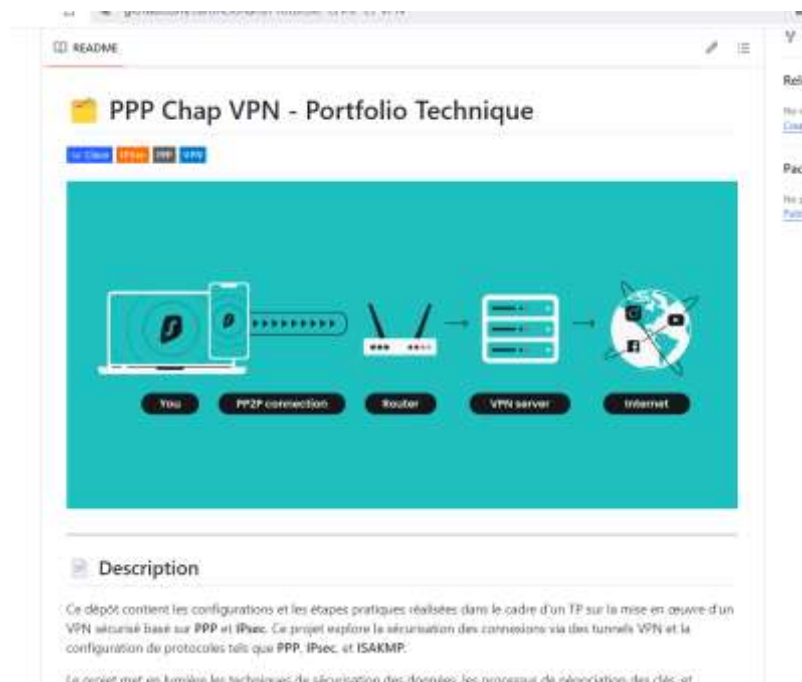
La configuration IPsec réalisée permet d'établir des tunnels VPN sécurisés adaptés aux besoins d'accès local et distant. En mettant en place des **tunnels IPsec** avec des algorithmes de chiffrement robustes tels que **AES 128 bits** et une intégrité via **SHA256**, nous avons garanti une protection optimale des données échangées. L'utilisation de l'authentification par **PSK** et **Xauth** renforce encore plus la sécurité en s'assurant que seuls les utilisateurs autorisés peuvent se connecter.

Grâce à l'activation du **support client IPsec** et à la configuration de pools d'adresses virtuelles, les utilisateurs mobiles peuvent facilement se connecter au réseau sans nécessiter d'intervention complexe. Les journaux de connexion confirment le bon fonctionnement des tunnels et montrent que les connexions des utilisateurs sont établies avec succès.

En somme, cette configuration IPsec offre une solution complète et sécurisée pour accéder au réseau, que ce soit pour des connexions fixes ou mobiles. Elle garantit confidentialité, intégrité et authentification des communications, répondant ainsi aux exigences de sécurité réseau en entreprise.

10. GITHUB :

Dans ce TP, un dépôt GitHub a été créé avec un fichier README bien structuré pour documenter et organiser les étapes et configurations du projet PPP Chap VPN. Les dossiers du projet ont été correctement gérés avec une séparation claire entre les fichiers de configuration (doc/) et les fichiers Packet Tracer ou GNS3 (pka/). Cette organisation permet une meilleure maintenance et compréhension du projet. Le README comprend une description détaillée, les étapes du TP, et un aperçu des technologies utilisées, ce qui garantit une documentation professionnelle et accessible.



 doc	05/01/2025 21:25	Dossier de fichiers
 pka	05/01/2025 21:26	Dossier de fichiers

La structure des dossiers du projet est bien organisée avec deux répertoires principaux :

- **doc/** : Contient la documentation et les fichiers associés, permettant une gestion claire et ordonnée des ressources textuelles.
- **pka/** : Regroupe les fichiers Packet Tracer ou GNS3 liés à la simulation et à la configuration réseau.

Cette séparation facilite la navigation et la maintenance du projet tout en garantissant une meilleure lisibilité des différentes composantes.

11. CONCLUSION :

Ce TP a permis de configurer et de valider une connexion sécurisée IPsec sur une architecture utilisant le protocole PPP. Les étapes principales ont inclus la mise en place des associations de sécurité (ISAKMP et IPsec), la définition des ACL pour spécifier le trafic à protéger, ainsi que la configuration des routes pour assurer la connectivité entre les réseaux.

Les tests ont démontré l'importance du débogage pour identifier et résoudre les erreurs, ainsi que la nécessité d'une configuration précise des adresses IP et des routes. Finalement, une fois les ajustements apportés, les paquets chiffrés et décapsulés ont validé le fonctionnement optimal d'IPsec, garantissant la confidentialité et l'intégrité des communications.

Ce TP met en évidence l'utilité d'IPsec dans un contexte de sécurisation réseau, particulièrement lorsqu'il est associé à PPP pour les connexions WAN.