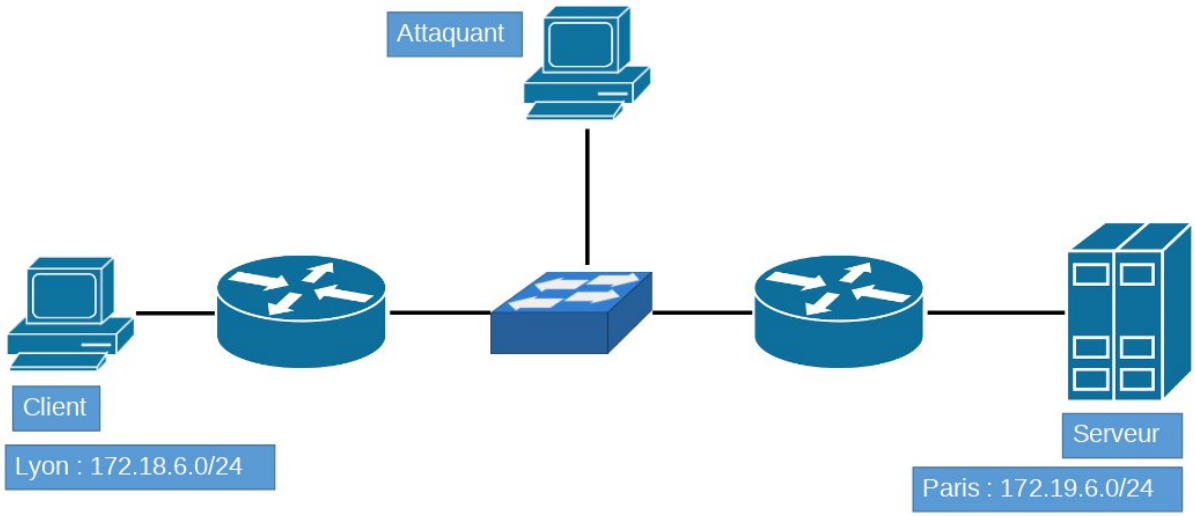
TP RS2P

Etape 0 – Mise en place de l'Infrastructure



- **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)***

  - Que devez vous faire pour mettre en place la connectivité ?

- **Configuration des Hôtes**
  - Que devez vous faire pour mettre en place la connectivité ?

Le ping vers le serveur FTP depuis le client doit fonctionner. Si non, revoir le réseau.

```
root@client-FTP:~# ping 172.19.6.100
PING 172.19.6.100 (172.19.6.100) 56(84) bytes of data.
64 bytes from 172.19.6.100: icmp_seq=1 ttl=62 time=3.85 ms
64 bytes from 172.19.6.100: icmp_seq=2 ttl=62 time=5.89 ms
64 bytes from 172.19.6.100: icmp_seq=3 ttl=62 time=3.03 ms
64 bytes from 172.19.6.100: icmp_seq=4 ttl=62 time=2.53 ms
^C
--- 172.19.6.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 2.526/3.826/5.894/1.284 ms
```

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

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)***

Depuis l'attaquant

- *Mise en place d'un port mirroring sur le switch pour récupérer le trafic.*
- *Avec wireshark observez le trafic.*
- Le client va à présent effectuer une requête ftp vers le serveur et nous allons lancer un tcpdump sur la machine pirate afin de capturer le trafic FTP et répetez les étapes préceentes entre le client et le serveur.
- Qu'observez vous ? Pourquoi ? 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

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)*

Attaquant

- Relancer la capture wireshark.

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 »

- Qu'observez vous ? Pourquoi ? Cette architecture est-elle sécurisée pour toutes les communications ? Que proposez vous ?

Etape 3 – Mise en place d'un tunnel VPN IPSec avec 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=192.168.152.77
    leftsubnet=172.18.6.0/24
    right=192.168.152.149
    rightsubnet=172.19.6.0/24
    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 "192.168.152.77 192.168.152.149 : 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=192.168.152.149
    leftsubnet=172.19.6.0/24
    right=192.168.152.77
    rightsubnet=172.18.6.0/24
    ike=aes256-sha1-modp1024!
    esp=aes256-sha1!
    aggressive=no
    keyingtries=%forever
    ikelifetime=28800s
    lifetime=3600s
    dpddelay=30s
    dpdtimeout=120s
    dpdaction=restart
```

- echo "192.168.152.149 192.168.152.77 : PSK \"vpn_key\"" | sudo tee -a /etc/ipsec.secrets ***(ajout de la clé PSK)***

Sur chaque Routeur

- *Vérifiez que les routes statiques sont toujours présentes :*

```
ubuntu@routeur-lyon:~$ sudo ip route
default via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.77 metric 100
10.10.10.10 via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.77 metric 100
10.10.10.11 via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.77 metric 100
10.247.0.0/25 dev docker0 proto kernel scope link src 10.247.0.1 linkdown
169.254.169.254 via 172.18.6.1 dev enp2s0 proto dhcp src 172.18.6.78 metric 100
169.254.169.254 via 192.168.152.3 dev enp1s0 proto dhcp src 192.168.152.77 metric 100
172.18.6.0/24 dev enp2s0 proto kernel scope link src 172.18.6.78 metric 100
172.18.6.1 dev enp2s0 proto dhcp scope link src 172.18.6.78 metric 100
172.19.6.0/24 via 192.168.152.149 dev enp1s0
```

```
ubuntu@routeur-paris:~$ sudo ip route
default via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.149 metric 100
10.10.10.10 via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.149 metric 100
10.10.10.11 via 192.168.152.1 dev enp1s0 proto dhcp src 192.168.152.149 metric 100
10.247.0.0/25 dev docker0 proto kernel scope link src 10.247.0.1 linkdown
169.254.169.254 via 172.19.6.1 dev enp2s0 proto dhcp src 172.19.6.14 metric 100
169.254.169.254 via 192.168.152.3 dev enp1s0 proto dhcp src 192.168.152.149 metric 100
172.18.6.0/24 via 192.168.152.77 dev enp1s0
172.19.6.0/24 dev enp2s0 proto kernel scope link src 172.19.6.14 metric 100
172.19.6.1 dev enp2s0 proto dhcp scope link src 172.19.6.14 metric 100
```

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

```
ubuntu@routeur-lyon:~$ sudo ipsec status
Security Associations (2 up, 0 connecting):
devgateway-to-prodgateway[2]: ESTABLISHED 8 seconds ago, 192.168.152.77[192.168.152.77]...192.168.152.149[192.168.152.149]
devgateway-to-prodgateway{1}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c1decd95_i c0978742_o
devgateway-to-prodgateway{1}:   172.18.6.0/24 === 172.19.6.0/24
```

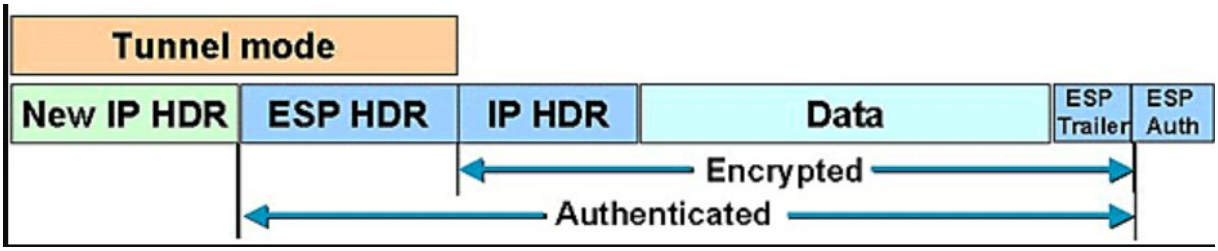
```
ubuntu@routeur-paris:~$ sudo ipsec status
Security Associations (2 up, 0 connecting):
devgateway-to-prodgateway[2]: ESTABLISHED 15 seconds ago, 192.168.152.149[192.168.152.149]...192.168.152.77[192.168.152.77]
devgateway-to-prodgateway{2}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: cb8c7c17_i c791e45a_o
devgateway-to-prodgateway{2}:   172.19.6.0/24 === 172.18.6.0/24
```

Vérifier que la communication est bien chiffrée entre le client et serveur. Lancer wireshark depuis le client, l'attaquant et le serveur.

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).

ESP protège complètement le datagramme d'origine car il s'agit désormais des données de contenu du nouveau paquet ESP. ESP, cependant, ne protège pas le nouvel en-tête IP. Les passerelles doivent utiliser ESP en mode tunnel.



Est ce qu'on peut quand même lancer des attaques ? De quels types ? (il y a plusieurs possibilités).

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 : hping3 - envoie des paquets TCP/IP (presque) arbitraires à des systèmes réseaux (man-linux-magique.net)]

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

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

```
ubuntu@serveur-paris:~$ sudo tcpdump port ftp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp1s0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:32:06.187863 IP 172.18.6.74.2054 > serveur-paris.univ-lyon1.fr.ftp: Flags [S], seq 59363
11:32:06.187937 IP serveur-paris.univ-lyon1.fr.ftp > 172.18.6.74.2054: Flags [S.], seq 3118
11:32:06.189623 IP 172.18.6.74.2054 > serveur-paris.univ-lyon1.fr.ftp: Flags [R], seq 59363
11:32:07.188116 IP 172.18.6.74.2055 > serveur-paris.univ-lyon1.fr.ftp: Flags [S], seq 13399
11:32:07.188190 IP serveur-paris.univ-lyon1.fr.ftp > 172.18.6.74.2055: Flags [S.], seq 4439
11:32:07.190325 IP 172.18.6.74.2055 > serveur-paris.univ-lyon1.fr.ftp: Flags [R], seq 13399
11:32:08.188174 IP 172.18.6.74.2056 > serveur-paris.univ-lyon1.fr.ftp: Flags [S], seq 14964
11:32:08.188246 IP serveur-paris.univ-lyon1.fr.ftp > 172.18.6.74.2056: Flags [S.], seq 1333
11:32:08.190162 IP 172.18.6.74.2056 > serveur-paris.univ-lyon1.fr.ftp: Flags [R], seq 14964
11:32:09.188409 IP 172.18.6.74.2057 > serveur-paris.univ-lyon1.fr.ftp: Flags [S], seq 14731
11:32:09.188505 IP serveur-paris.univ-lyon1.fr.ftp > 172.18.6.74.2057: Flags [S.], seq 4288
11:32:09.190851 IP 172.18.6.74.2057 > serveur-paris.univ-lyon1.fr.ftp: Flags [R], seq 14731
11:32:10.188374 IP 172.18.6.74.2058 > serveur-paris.univ-lyon1.fr.ftp: Flags [S], seq 19292
11:32:10.188450 IP serveur-paris.univ-lyon1.fr.ftp > 172.18.6.74.2058: Flags [S.], seq 3071
11:32:10.190134 IP 172.18.6.74.2058 > serveur-paris.univ-lyon1.fr.ftp: Flags [R], seq 19292
```

- *netstat -tunap* (pour voir les connexions actives ------✉ à faire pour la commande avancée)

```
ubuntu@serveur-paris:~$ netstat -tunap
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0    576 172.19.6.186:22         172.19.6.14:46004       ESTABLISHED -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::21                   :::*                    LISTEN      -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23505       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23501       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23518       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23517       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23378       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23389       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23385       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23515       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23504       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23377       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23516       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23514       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23388       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23381       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23379       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23506       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23510       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23382       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23509       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23383       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23502       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23386       SYN_RECV    -
tcp6       0      0 172.19.6.186:21         172.18.6.74:23642       SYN_RECV    -
```

Question ?

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