




Atelier 1 PKI Guillaume Sanchez

Environnement :

3 containers proxmox mis, en place à l'aide de "Atelier 1 - Préparation.pdf"

 206 (alpine-RSX112)	- pki-ca IP: 192.168.1.232 ID 206
 207 (alpine-RSX112-web-server)	- web-server IP : 192.168.1.233 ID 207
 208 (alpine-RSX112-Client-Admin)	- client-admin IP : 192.168.1.234 ID 208

Construction de la PKI Minimaliste :

Après la mise en place des trois serveurs et de leur configuration, on a pu générer les premiers certificats avec le script d'automatisation "issue-cert.sh" :

```
alpine-RSX112:~/mini-ca$ ./issue-cert.sh web-server
Using configuration from openssl-minimal.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'web-server'
organizationName :ASN.1 12:'Groupe X'
countryName     :PRINTABLE:'FR'
Certificate is to be certified until Jun 25 07:22:16 2026 GMT (365 days)

Write out database with 1 new entries
Database updated
Certificat émis: certs/web-server.crt.gz
```

Ce certificat a été placé sur le serveur web :

```
alpine-RSX112-web-server:~$ scp pki-ca:/home/pkilab/mini-ca/certs/web-server.crt.gz ~/ssl/
pkilab@192.168.1.232's password:
web-server.crt.gz                                100% 2540      4.6MB/s   00:00
alpine-RSX112-web-server:~$ scp pki-ca:/home/pkilab/mini-ca/private/web-server.key.gz ~/ssl/
pkilab@192.168.1.232's password:
web-server.key.gz                                100% 1331      2.7MB/s   00:00
```

Et la configuration de nginx a été réalisé avec ce certificat :

```
alpine-RSX112-web-server:~/ssl$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Problème rencontré pendant la configuration de nginx:

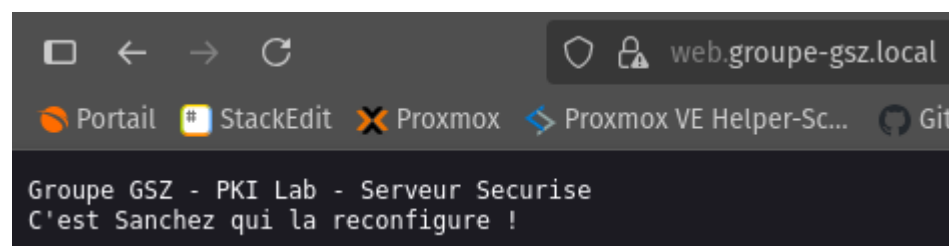
J'ai rencontré plusieurs problèmes pendant la configuration, vous demandez dans vos consignes de créer "ssl.conf" dans "/etc/nginx/conf.d", mais le répertoire par défaut serait plutôt "http.d" car "Include /etc/nginx/conf.d/*.conf" ne se trouve pas dans la balise "http" dans le fichier nginx.conf. J'ai déplacé la ligne dans la balise et le problème a été résolu.

Egalement, dans le fichier **"ssl.conf"**, la ligne **"ssl_session_cache shared:SSL:1m;"** ne plait pas à nginx car il utilise une zone mémoire partagée nommée "SSL" pour stocker les sessions TLS et nginx ne permet pas de redéfinir la même zone partagée (shared:SSL) avec une taille différente. J'ai simplement commenté la ligne dans ssl.conf et le problème fu résolu.

Une fois la configuration terminé et testé, j'ai pu démarrer nginx :

```
alpine-RSX112-web-server:~/ssl$ sudo rc-service nginx start
* Starting nginx ... [ ok ]
```

Vérifier que ma page web s'affichais bien :



Et que le bon certificat était utilisé :

Certificat	
web-server	
Nom du sujet	
Pays	FR
Organisation	Groupe X
Nom courant	web-server
Nom de l'émetteur	
Nom courant	Groupe CA
Organisation	PKI Lab
Pays	FR
Validité	
Pas avant	Wed, 25 Jun 2025 07:22:16 GMT
Pas après	Thu, 25 Jun 2026 07:22:16 GMT

Scénarios Adaptés aux Contraintes

J'ai mis en place le script "incident-response-lite.sh" auquel j'ai réalisé une multitude de modification pour le faire fonctionner. Ce script permet de lister la liste des certificats en cours afin de pourvoir les révoquer et les régénérer. Il génère également un petit compte rendu de l'incident avec la date et l'heure.

Voici le script réécrit :

```
#!/bin/bash

echo "=== INCIDENT RESPONSE TOOLKIT (Lite) ==="
echo "Optimisé pour environnements contraints"
echo ""

# 1. Isolation réseau (simulée)
echo "[1] Isolation réseau du serveur compromis"
echo "      # iptables -A INPUT -s 192.168.1.233 -j DROP"
echo "      (Non exécuté en lab)"

# 2. Collecte d'informations minimale
echo -e "\n[2] Collecte d'informations"
mkdir -p ../incident-$(date +%Y%m%d)
cd ../incident-$(date +%Y%m%d)

# Inventaire des certificats (version légère)
echo -e "\n[*] Inventaire des certificats actifs:"
ssh pki-ca "cd ~/mini-ca && cat index.txt" > cert-inventory.txt
cat cert-inventory.txt

# 3. Révocation
echo -e "\n[3] Révocation du certificat compromis"
read -p "Entrer le numéro de série à révoquer: " SERIAL

ssh pki-ca "cd ~/mini-ca && \
openssl ca -config openssl-minimal.cnf -revoke certs/$SERIAL.pem && \
openssl ca -config openssl-minimal.cnf -gencrl -out crl/ca.crl && \
echo 'CRL générée: ' \$(wc -c < crl/ca.crl) 'bytes'"

# 4. Notification (version simple)
echo -e "\n[4] Notifications"

echo "ALERTE SÉCURITÉ - Groupe GSZ" >> notification.txt
echo "===== " >> notification.txt
echo "Date: $(date)" >> notification.txt
echo "Certificat révoqué: $SERIAL" >> notification.txt
echo "Action: Mettre à jour la CRL" >> notification.txt
echo "Contact: pkilab@groupeGSZ" >> notification.txt

echo "Notification créée: notification.txt"

# 5. Surveillance mémoire pendant l'incident
echo -e "\n[5] Impact sur les ressources:"
free -h
df -h ~/
```

J'ai simplement réécrit le passage `cd` la connexion au serveur "pki-ca" avec le passage des commandes, et j'ai opté pour des "echo ... >> notification.txt" à la place du `cat`.

Voici le résultat après le lancement du scripts, il y a sur cette capture d'écrans une erreur **"ERROR:Already revoked, serial number 01"** , car j'avais déjà révoqué ce certificat lors de test. Cela m'a généré un nouveau certificat avec comme série "02" :

```
alpine-RSX112-Client-Admin:~/pki-lab/scripts$ ./incident-response-lite.sh
=== INCIDENT RESPONSE TOOLKIT (Lite) ===
Optimisé pour environnements contraints

[1] Isolation réseau du serveur compromis
# iptables -A INPUT -s 192.168.1.233 -j DROP
(Non exécuté en lab)

[2] Collecte d'informations

[*] Inventaire des certificats actifs:
R      260625072216Z    250625085754Z    01      unknown /C=FR/O=Groupe X/CN=web-server

[3] Révocation du certificat compromis
Entrer le numéro de série à révoquer: 01
Using configuration from openssl-minimal.cnf
ERROR:Already revoked, serial number 01

[4] Notifications
Notification créée: notification.txt

[5] Impact sur les ressources:
      total      used      free      shared  buff/cache  available
Mem:    15.4G    11.3G     1.9G    129.8M     2.2G     377.8M
Swap:     8.0G     30.0M     8.0G
Filesystem
/dev/loop11      5.8G    73.0M     5.4G    1% /
```

Monitoring des Ressources

Mise en place du script monitor-pki.sh :

```
=== PKI Monitor - Groupe GSZ ===
Time: Wed Jun 25 09:56:26 UTC 2025

[Conteneurs]
192.168.1.232: ✓ UP
192.168.1.233: ✓ UP
192.168.1.234: ✓ UP

[CA Resources]
Mem:    15.4G    11.3G     1.9G    129.8M     2.2G     248.7M

[Certificates]
Active certificates: 1

[Disk Usage]
Used: 73.0M / 5.8G (1%)
```

Ce script nous permet de voir que nos contener Proxmox sont bien en route et le nombre de ressource qui sont disponible dessus.

Optimisations Finales

J'ai réalisé le script d'optimisation final sur le serveur "Client Admin". J'ai utilisé le protocole ssh pour lancer les commandes à distance sur les serveurs "pki-ca" et "web-server". J'ai également rajouté le lancement du script "incident-response-liste.sh" afin de révoquer les potentiel problème de certificats.

Voici mon script après les modifications :

```
#!/bin/bash

echo "=== Optimisation PKI ==="

# 1. Compression des logs
echo "=== Compression des logs ==="
ssh pki-ca 'find ~/mini-ca -name "*.log" -exec gzip {} \;'

# 2. Nettoyage des fichiers temporaires
echo "=== Nettoyage des fichiers temporaires ==="
ssh pki-ca rm -f ~/mini-ca/csr/*
ssh pki-ca rm -f ~/mini-ca/*.old

# 3. Rotation des certificats expirés
echo "=== Rotation des certificats expirés ==="
~/pki-lab/scripts/incident-response-lite.sh

# 4. Optimisation mémoire Nginx
echo "=== Optimisation mémoire Nginx ==="
ssh web-server sudo killall -HUP nginx

echo "Optimisation terminée!"
```

Voici le résultat de ce script :

```
echo -e "\n[2] Collecte d'informations"
alpine-RSX112-Client-Admin:~/pki-lab/scripts$ ./optimize-pki.sh
=== Optimisation PKI ===
=== Compression des logs ===
=== Nettoyage des fichiers temporaires ===
=== Rotation des certificats expirés ===
=== INCIDENT RESPONSE TOOLKIT (Lite) ===
Optimisé pour environnements contraints

[1] Isolation réseau du serveur compromis
# iptables -A INPUT -s 192.168.1.233 -j DROP
(Non exécuté en lab)

[2] Collecte d'informations

[*] Inventaire des certificats actifs:
R      260625072216Z  250625085754Z  01      unknown /C=FR/O=Groupe X/CN=web-server

[3] Révocation du certificat compromis
Entrer le numéro de série à révoquer: n
Using configuration from openssl-minimal.cnf
Could not open file or uri for loading certificate to be revoked from certs/n.pem: No such file or directory

[4] Notifications
Notification créée: notification.txt

[5] Impact sur les ressources:
      total      used      free      shared  buff/cache  available
Mem:   15.4G     11.3G      1.9G     129.8M      2.2G     377.2M
Swap:   8.0G      30.0M      8.0G
Filesystem      Size      Used Available Use% Mounted on
/dev/loop11     5.8G      73.0M      5.4G   1% /
=== Optimisation mémoire Nginx ===
Optimisation terminée!
```

Débriefing et Évaluation

Questions de Réflexion

1. Contraintes et Solutions

a. "Comment les limitations de ressources ont-elles influencé vos choix ?"

Je n'ai pas eu de limitation de ressources car j'ai réalisé le projet depuis mon environnement. Les containers qui eux, ont eu des ressources limitées ne m'ont pas gêné non plus pour réaliser le projet. Il y avait largement de quoi créer plusieurs certificats et lancer les différents scripts sans trop de mal.

b. "Quelles optimisations supplémentaires pourriez-vous implémenter ?"

Une vérification automatique de la validité des certificats selon leur date afi

2. Scalabilité

a. "Cette infrastructure pourrait-elle supporter 100 certificats ?"

Je ne suis pas sûr, mais je pense que oui. Même en termes de capacité de stockage limité, il y a quand même 5.8Go de disponible. Un certificat ne prend qu'une dizaine de Mo.

b. "Comment automatiser davantage sans augmenter les ressources ?"

On peut utiliser des crontabs pour réaliser de la surveillance automatique qui utiliserait les scripts déjà mis en place, nettoierait les logs trop vieux et générerai des rapports de situation régulièrement. On pourrait également automatiser la vérification et la révocation de certificat de manière automatique également.

3. Sécurité vs Performance

a. "Quels compromis de sécurité avez-vous dû faire ?"

L'environnement actuel est ouvert sans aucune restriction ou firewall.

b. "Sont-ils acceptables en production ?"

Non en production, les accès au réseau ou au Vlan serait limité et un firewall serait également mis en place.