Titre : Sudo, Pam, Fail2ban et Ufw
Contexte : Activité professionnelle N°15

Compétences du bloc 1 :

- **Travailler en mode projet**
  - Analyser les objectifs et les modalités d'organisation d'un projet
  - Planifier les activités

Compétences du bloc 2 :

- Installer, tester et déployer une solution d'infrastructure réseau
- Rédiger ou mettre à jour la documentation technique et utilisateur d'une solution d'infrastructure
- Tester l'intégration et l'acceptation d'une solution d'infrastructure

Compétences du bloc 3 :

- **Sécuriser les équipements et les usages des utilisateurs**
  - Identifier les menaces et mettre en œuvre les défenses appropriées
  - Gérer les accès et les privilèges appropriés
  - Vérifier l'efficacité de la protection

Contraintes :

- Créez le GANTT pour cette nouvelle AP
- Complétez votre schéma réseau et votre document de synthèse sur les VLANs et VMs créés dans votre environnement.
- Rendre votre document de synthèse avant vendredi minuit au plus tard.

Prérequis :

- Serveur OpenVPN et client VPN fonctionnels.
- Serveur de mail et client de messagerie fonctionnels.
- Serveur central de log fonctionnel
- Si nécessaire, désactivez l'authentification SSH par clef publique sur les VMs utilisées dans cette AP.

1] Contexte

En tant que technicien, vous avez déjà amélioré la sécurité de votre environnement Menuimétal mais le RSSI (Responsable de la sécurité des systèmes d'information) souhaite que vous poursuivez cette tâche.

Il souhaite notamment que vous :

- affiniez les pouvoirs d'administration de vos administrateurs,
- renforciez le système d'authentification pour le service VPN,
- mettiez en place le système d'alerte par mail pour l'IDS/IPS Fail2ban,
- implémentiez des contre-mesures Fail2ban pour le service OpenVPN,
- installiez et configuriez un système dit de pare-feu.

Toutes les configurations que vous allez réaliser, si vous vous trompez, peuvent compromettre l'accès aux VMs utilisées dans cette AP, vous ferez donc OBLIGATOIREMENT une SAUVEGARDE avant toute intervention sur ces dernières.

2] sudo ou « l'affinage » des privilèges

Documentations utiles :

- https://www.it-connect.fr/commande-sudo-comment-configurer-sudoers-sous-linux/

Questions :

- Qu'est-ce que le concept « de moindre privilège » ?
- Si on utilise la commande **Sudo** (à installer sous Debian), de quel groupe par défaut faut-il faire partie afin d'acquérir des privilèges d'administration ?
- Sous Linux, comment vérifier notre appartenance à un groupe d'utilisateurs ?
- C'est quoi le fichier **sudoers** ?
- C'est quoi la commande **visudo** ? Quel est son avantage ?

Objectifs attendus et contraintes :

- Un administrateur dédié (**adminvpn**) doit être le seul à pouvoir administrer le service **openvpn** par l'utilisation du script **easyrsa**, sans avoir à saisir son mot de passe.

- Un administrateur dédié (**adminutil**) doit être le seul à pouvoir administrer les utilisateurs systèmes créés sur le serveur Linux utilisé pour Openvpn.

- Les logs générés par **sudo** (fichier **/var/log/auth.log** à condition d'avoir installé **rsyslog**) devront être redirigés vers votre serveur central **rsyslog**. Vous pouvez également consulter en local les logs avec la commande **journalctl _COMM=sudo**).

3] Module PAM ou l'authentification plus robuste

Documentations utiles :

- https://www.cnil.fr/fr/mots-de-passe-une-nouvelle-recommandation-pour-maitriser-sa-securite
- https://www.economie.gouv.fr/particuliers/creer-mot-passe-securise
- https://cyber.gouv.fr/publications/recommandations-relatives-lauthentification-multifacteur-et-aux-mots-de-passe
- https://blog.stephane-robert.info/docs/securiser/durcissement/pam/

Questions :

- Quel est l'intérêt des modules PAM ?

Objectifs attendus et contraintes :

- Chaque administrateur (adminutil et adminvpn) du serveur VPN doit pouvoir changer son mot de passe pour renforcer la contrainte de complexité et donc la sécurité :
    - Vous installerez le module **libpam-pwquality**.
    - En utilisant cette documentation (https://debian-facile.org/doc:securite:passwd:libpam-pwquality), vous configurez le fichier **/etc/pam.d/common-password** pour respecter les règles de sécurité suivantes :
        - Longueur minimum de 10 caractères,
        - utilisation des minuscules, majuscules, chiffres et caractères spéciaux,
        - trois essais maximum.

- Le RSSI souhaite également appliquer des restrictions horaires aux administrateurs du serveur VPN :
    - En utilisant cette documentation (https://doc.ubuntu-fr.org/tutoriel/restrictions_horaires), vous configurerez les fichiers **/etc/pam.d/common-auth** et **/etc/security/time.conf** pour autoriser l'accès :
        - Les jours ouvrés donc du Lundi au Vendredi,
        - de 08H00 à 17H00.

4] Fail2ban et OpenVPN

Existant :

Pour rappel, Fail2ban est déjà en place sur l'un de vos serveurs pour le service SSH. Les paquets nécessaires à sa gestion étaient notamment **fail2ban** et **iptables**. A l'aide des commandes suivantes :

- **fail2ban-client status**
- **fail2ban-client status sshd**
- **iptable -L -n -v**

Vous devez donc être capable de démontrer qu'une adresse IP peut-être bannie lorsque trois essais infructueux d'identifiants ont été réalisés. De plus, une notification par mail doit être transmise à un responsable de votre choix.

Questions :

- C'est quoi l'authentification multi facteur ?

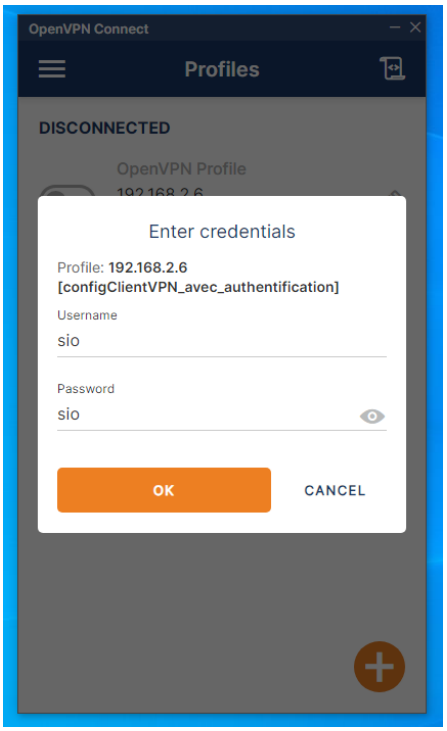
Objectifs attendus et contraintes :

- Le RSSI souhaite maintenant, que vous renforciez l'authentification des utilisateurs pour le service OpenVPN.

  En effet, en plus d'une authentification par certificat comme vous l'avez déjà réalisée, seuls les utilisateurs (non administrateur) référencés sur le serveur VPN doivent pouvoir se connecter à l'aide de leurs identifiants habituels.

  L'utilisation des modules PAM est requise.

  Voici ce qui se passera côté client sur la fenêtre du client **OpenVPN connect** :



A l'aide de ces documentations :

- https://www.ksh-linux.info/reseaux/vpn/openvpn-config-tun
- https://openvpn.net/community-docs/using-alternative-authentication-methods.html
- https://www.charlesreid1.com/wiki/Ubuntu/OpenVPN_Server?utm_source=chatgpt.com

Vous devez paramétrer côté serveur l'authentification des utilisateurs Linux :

- Créez un nouveau profil **PAM** nommé **openvpn** (fichier à créer **/etc/pam.d/openvpn**) qui utilisera l'authentification Linux à l'aide du fichier habituel **shadow**.
- Modifiez la configuration de votre fichier **server.conf** pour utiliser le plugin **/usr/lib/openvpn/plugins/openvpn-plugin-auth-pam.so** avec le nouveau profil **PAM openvpn**.
- Vous penserez à désactiver la configuration **chroot** mise en place dans l'AP précédente car elle pose des problèmes pour cette nouvelle contrainte.
- Vous ajouterez également un fichier de suivi des adresses affectées dans le cadre des connexions VPN réussies (option **ifconfig-pool-persist**).

Vous devez paramétrer côté client l'authentification des utilisateurs Linux :

- Ajouter dans votre fichier de configuration l'option **auth-user-pass**.

Voici ce qui est affiché par la commande **tail -f /var/log/openvpn.log** :

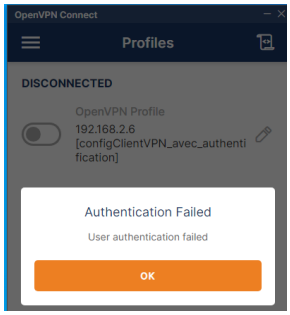
```
2024-12-08 19:06:18 us=373181 192.168.2.1:58251 [sio] Peer Connection Initiated with [AF_INET]192.168.2.1:58251
2024-12-08 19:06:18 us=374115 sio/192.168.2.1:58251 MULTI_sva: pool returned IPv4=10.8.0.10, IPv6=(Not enabled)
2024-12-08 19:06:18 us=375052 sio/192.168.2.1:58251 MULTI: Learn: 10.8.0.10 -> sio/192.168.2.1:58251
2024-12-08 19:06:18 us=375932 sio/192.168.2.1:58251 MULTI: primary virtual IP for sio/192.168.2.1:58251: 10.8.0.10
2024-12-08 19:06:18 us=375990 sio/192.168.2.1:58251 Data Channel MTU parms [ mss_fix:1400 max_frag:0 tun_mtu:1500 tun_max_mtu:1600 headroom:136 payload:1768 tailroom:562 ET:0 ]
2024-12-08 19:06:18 us=376057 sio/192.168.2.1:58251 Outgoing dynamic tls-crypt: Cipher 'AES-256-CTR' initialized with 256 bit key
2024-12-08 19:06:18 us=376077 sio/192.168.2.1:58251 Outgoing dynamic tls-crypt: Using 256 bit message hash 'SHA256' for HMAC authentication
2024-12-08 19:06:18 us=376085 sio/192.168.2.1:58251 Incoming dynamic tls-crypt: Cipher 'AES-256-CTR' initialized with 256 bit key
2024-12-08 19:06:18 us=376097 sio/192.168.2.1:58251 Incoming dynamic tls-crypt: Using 256 bit message hash 'SHA256' for HMAC authentication
2024-12-08 19:06:18 us=376124 sio/192.168.2.1:58251 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-12-08 19:06:18 us=376158 sio/192.168.2.1:58251 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-12-08 19:06:18 us=376194 sio/192.168.2.1:58251 SENT CONTROL [sio]: 'PUSH_REPLY,iroute 192.168.2.0 255.255.255.0,route 10.8.0.1 255.255.255.255,dhcp-option DNS 192.168.2.9,redirect-gateway def1,route 10.8.0.0 255.255.255.0,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.10 10.8.0.9,peer-id 0,cipher AES-256-GCM,protocol-flags cc-exit tls-ekm dyn-tls-crypt,tun-mtu 1500' (status=1)
```

Voici ce qui est affiché par la commande **cat ipp.txt** :

```
root@debian:/etc/openvpn# cat ipp.txt
testuser,10.8.0.4,
sio,10.8.0.8,
```

Et voici ce qui se passe lorsque de mauvais identifiants sont détectés :

```
2024-12-08 19:14:04 us=454312 PLUGIN_AUTH-PAM: BACKGROUND: received command code: 0
2024-12-08 19:14:04 us=454342 PLUGIN_AUTH-PAM: BACKGROUND: USER: sio
2024-12-08 19:14:04 us=454347 PLUGIN_AUTH-PAM: BACKGROUND: REMOTE: 192.168.2.1
2024-12-08 19:14:04 us=455541 PLUGIN_AUTH-PAM: BACKGROUND: my_conv[0] query='Password: ' style=1
2024-12-08 19:14:04 us=485153 PLUGIN_AUTH-PAM: BACKGROUND: user 'sio' failed to authenticate: Authentication failure
2024-12-08 19:14:04 us=486246 192.168.2.1:51337 PLUGIN_CALL: POST /usr/lib/openvpn/openvpn-plugin-auth-pam.so/PLUGIN_AUTH_USER_PASS_VERIFY status=1
2024-12-08 19:14:04 us=488007 192.168.2.1:51337 PLUGIN_CALL: plugin function PLUGIN_AUTH_USER_PASS_VERIFY failed with status 1: /usr/lib/openvpn/openvpn-plugin-auth-pam.so
2024-12-08 19:14:04 us=489778 192.168.2.1:51337 TLS Auth Error: Auth Username/Password verification failed for peer
2024-12-08 19:14:04 us=490658 192.168.2.1:51337 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2024-12-08 19:14:04 us=491482 192.168.2.1:51337 TLS: tls_multi_process: initial untrusted session promoted to semi-trusted
2024-12-08 19:14:04 us=492199 192.168.2.1:51337 Delayed exit in 5 seconds
2024-12-08 19:14:04 us=493094 192.168.2.1:51337 SENT CONTROL [UNDEF]: 'AUTH_FAILED' (status=1)
2024-12-08 19:14:04 us=493135 192.168.2.1:51337 SENT CONTROL [CltVPN]: 'AUTH_FAILED' (status=1)
2024-12-08 19:14:04 us=501100 192.168.2.1:51337 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048 bit RSA, signature: RSA-SHA256
2024-12-08 19:14:04 us=501156 192.168.2.1:51337 [CltVPN] Peer Connection Initiated with [AF_INET]192.168.2.1:51337
2024-12-08 19:14:06 us=905046 read UDPv4 [ECONNREFUSED]: Connection refused (fd=7,code=111)
2024-12-08 19:14:09 us=269444 192.168.2.1:51337 SIGTERM[soft,delayed-exit] received, client-instance exiting
```



La connexion pour l'utilisateur sio depuis le poste 192.168.2.1 est refusée

Par défaut, il n'existe aucun filtre fail2ban pour l'application OpenVPN (répertoire des filtres : **/etc/fail2ban/filter.d/**), vous allez donc devoir en créer un nouveau.

Documentations utiles :

- https://blog.louro.fr/securite/creer-un-filtre-fail2ban/
- https://regex101.com/ + https://www.arthurperret.fr/cours/expressions-regulieres.html
- https://www.it-connect.fr/filtres-et-actions-personnalises-dans-fail2ban/

Voici la procédure à adopter :

- Créez dans le répertoire **/etc/fail2ban/filter.d/** un nouveau fichier (extension en **.conf**) contenant le filtre adéquat qui permettra de détecter les erreurs d'authentification. Le patron à adopter est :

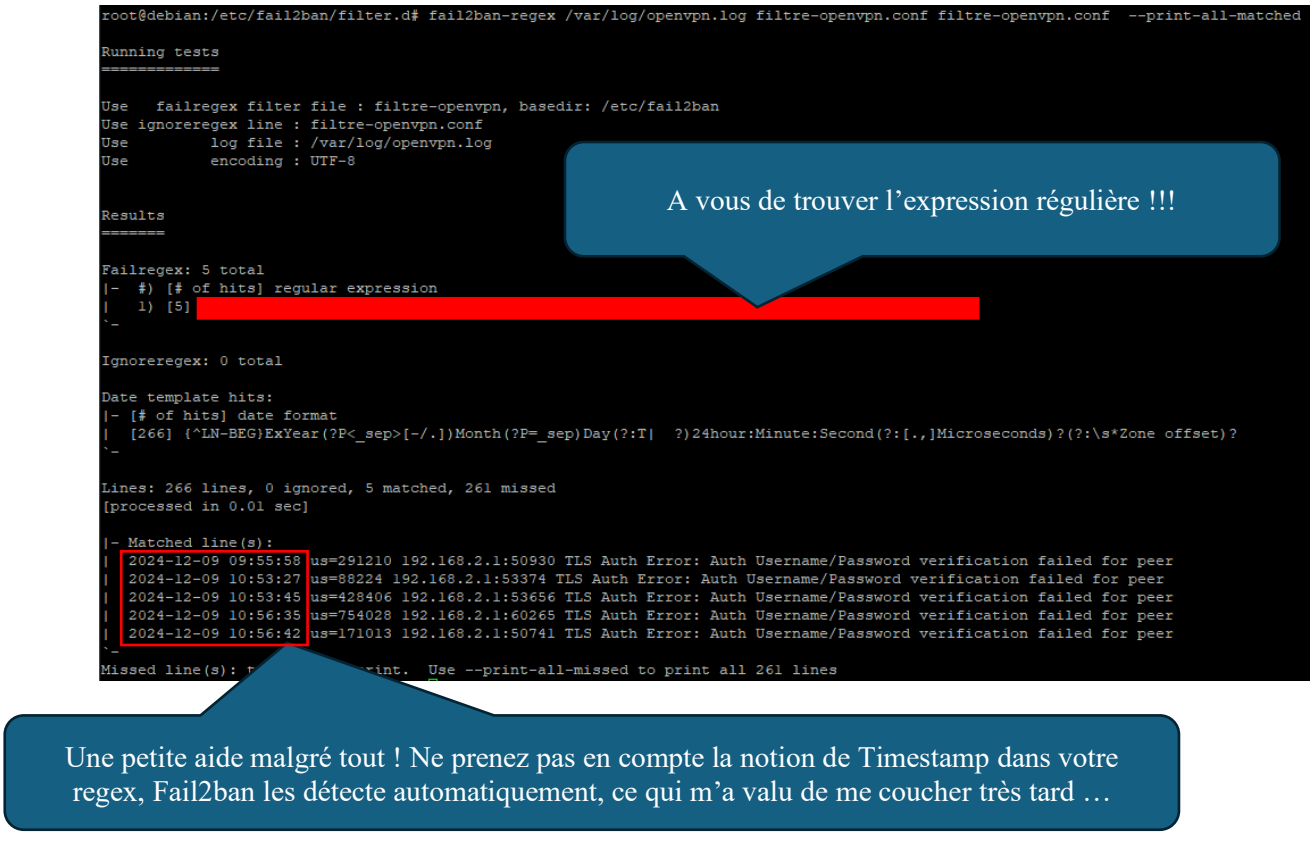
```
[Definition]

failregex =       Indiquer l'expression régulière à utiliser pour
                  trouver les authentifications erronées.

ignoreregex =     Indiquer l'expression régulière qui permettra
                  d'exclure certaines lignes si nécessaire.
```

Vous pouvez ensuite tester en mode console ce filtre avec la commande :

```
fail2ban-regex      NomFichierdesLogs.log      NomFichierduFiltre.conf
NomFichierduFiltre.conf --print-all-matched
```

Ce n'est pas une erreur qu'il y ait deux fois de suite « `NomFichierduFiltre.conf` », la première étant pour la directive « failregex » et la seconde pour « ignoreregex ».

S'il y a des occurrences correspondantes, elles seront affichées. Voici le résultat lors de mes tests :



- Créez dans le fichier **/etc/fail2ban/jail.local** (copie du fichier **jail.conf**), une nouvelle prison (jail in English) pour votre service VPN que vous nommerez **openvpn** :
  - Activez la surveillance.
  - Indiquer le port réseau d'OpenVPN.
  - Indiquer le fichier de log utilisé
  - Indiquer le filtre créé pour OpenVPN
  - Bannissez les IP au bout de deux essais infructueux pendant 120 secondes.

- Redémarrez le service Fail2ban puis vérifiez l'état de la prison pour OpenVPN (commandes **fail2ban-client status** et **fail2ban-client status openvpn**) voir **tail -f /var/log/auth.log**

- Depuis une autre VM équipée d'un client OpenVPN Connect, tentez une connexion avec un mauvais mot de passe puis vérifiez que son adresse IP est bannie. On peut utiliser les commandes précédentes voire **iptables -L -n -v**

- Faire en sorte qu'un mail soit transmis au responsable du service VPN afin de l'avertir du bannissement d'une IP.

- Vérifiez les logs du service Fail2ban avec **journalctl** ou **tail -f /var/log/fail2ban**

5] UFW ou le pare-feu « facile »

UFW est un nouvel outil de configuration simplifié en ligne de commande de **Netfilter**, qui donne une alternative à l'outil i**ptables**.

Deux possibilité de mise en place s'offrent à nous pour le dispositif de pare-feu :
1. Sur les postes client ou serveur donc en local,
2. sur le routeur lui-même puisqu'il fait la jonction entre tous les réseaux IP.

Documentations utiles :
- https://doc.ubuntu-fr.org/ufw
- https://www.digitalocean.com/community/tutorials/ufw-essentials-common-firewall-rules-and-commands-fr
- https://blog.stephane-robert.info/docs/securiser/reseaux/ufw/
- https://www.it-connect.fr/configurer-un-pare-feu-local-sous-debian-11-avec-ufw/
- https://grafikart.fr/tutoriels/ufw-696
- https://www.zdnet.fr/pratique/linux-comment-utiliser-uncomplicated-firewall-39944706.htm

Vos premières configurations sont à réaliser sur votre serveur de **mail**.
Pour commencer, vous devez :
- Installer UFW,
- Activez les logs (consultable par la commande : **tail -f /var/log/ufw.log**) ; le paquet **rsyslog** doit donc être installé,
- vérifiez les règles par défaut (il y en a trois),
- modifier la règle par défaut sur le trafic sortant en l'interdisant,
- désactivez les règles IPv6,
- permettre l'accès SSH, activez le pare-feu UFW, vérifiez alors le statut de vos règles et réalisez un test pour valider votre configuration (avant et après votre configuration).

Le « ping » reste possible malgré l'activation du pare-feu.

- Sans exécuter les commandes, trouvez les commandes qui permettent de désactiver le pare-feu et/ou de le réinitialiser (reset in English).
- Pour votre serveur de mail, vous mettrez en place les règles qui permettront de ne rendre accessible que les services réseaux mis en place :
    - service d'envoi de mail : SMTP
    - service de réception de mail IMAP

Réalisez des tests pour valider toutes ces configurations, notamment par la lecture des logs.