Hardening

Login SSH :

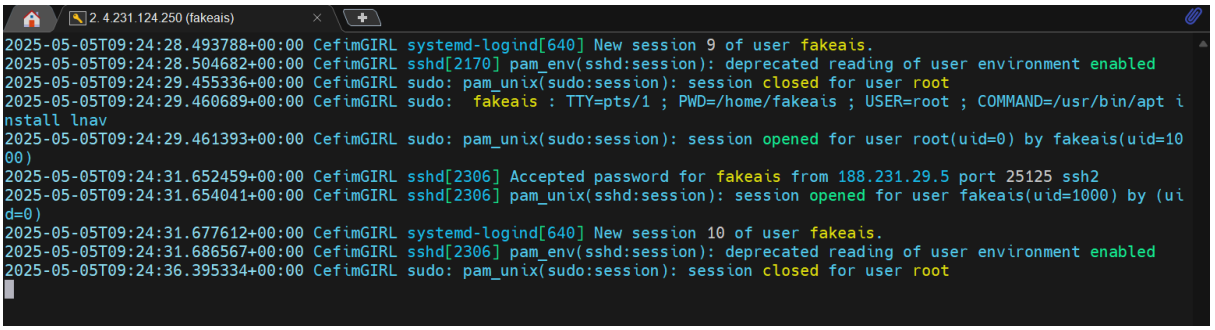
IP Address : **4.231.124.250**

User : **fakeais**
Password : 0warG1VHFJt3h4fABZZL8

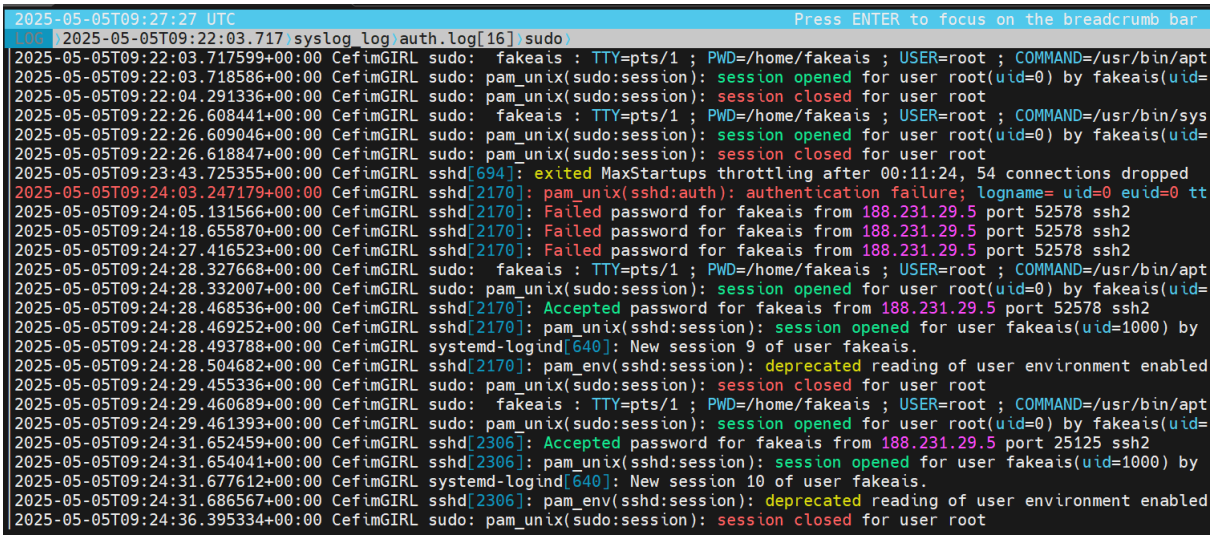
SSH Observation

Set Up - Avec les logs - auth.log

```
tail -f /var/log/auth.log | ccze
```




```
lnav /var/log/auth.log
```



Filtrage

Synthèses des tentatives SSH en échec sur les 24 dernières heures

```
fakeais@CefimGIRL:~$ awk -v since="$(date --date='24 hours ago' '+%Y-%m-%dT%H:%M:%S')" '
  $0 ~ /Failed password/ && $1 >= since { count++ }
  END { print count }
' /var/log/auth.log
3
fakeais@CefimGIRL:~$
```

SSH Enrichissement

installation WHOIS:

```
sudo apt install whois
```

installation de Tshark:

```
sudo apt install tshark
```

Configuration Python

```
sudo apt install python3-pip python3-venv
python3 -m venv ~/geoip_env
source ~/geoip_env/bin/activate
pip install pandas geoip2 folium
```

```
# Définir un compte de connexion pour le service SSH
AllowUsers fakeais
```

```
Port 56789
```

```
HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

KexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
```

Affichage des tentatives en échec par IP :

```
echo -e "Occurrences   | IP Address"
echo "------------------------------"
awk '
/sshd\[.*\]: Invalid user|Failed password/ {
    if (match($0, /from ([0-9]+\.[0-9]+\.[0-9]+\
        ip_count[arr[1]]++
    }
}
END {
    for (ip in ip_count) {
        printf "%-13s | %-15s\n", ip_count[ip], ip
    }
}' /var/log/auth.log | sort -nr
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash try_auth_by_ip.sh
Occurrences   | IP Address
------------------------------
3             | 188.231.29.5
2             | 71.176.69.9
2             | 188.231.29.10
```

Affichage des usernames utilisées :

```
echo -e "Occurrences   | Username" && \
echo "----------------------------------------
awk '/sshd\[.*\]: Invalid user/ {count[$6]++} E
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash try_username.sh
Occurrences   | Username
----------------------------------------
1             | ubnt
```

Affichage des usernames utilisées par IP :

```
echo -e "Occurrences   | IP Address      | User
echo "----------------------------------------
awk '
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash try_username_by_ip.sh
Occurrences   | IP Address      | Usernames
------------------------------------------------------------
1             | 71.176.69.9     | ubnt
```

```
/sshd\[.*\]: Invalid user/ {
    if (match($0, /Invalid user ([^ ]+) from ([0-9
        ip_count[arr[2]]++
        user_attempts[arr[2], arr[1]]++
        if (!seen[arr[2], arr[1]]) {
            user_list[arr[2]] = user_list[arr[2]] (use
            seen[arr[2], arr[1]] = 1
        }
    }
}
END {
    for (ip in ip_count) {
        printf "%-13s | %-15s | %s\n", ip_count[ip
    }
}' /var/log/auth.log | sort -nr
```

Synthèse des dernières 24h :

```
awk -v since="$(date --date='24 hours ago' '
  $0 ~ /Failed password/ && $1 >= since { cou
  END { print count }
' /var/log/auth.log
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash resume_try_auth.sh
6
```

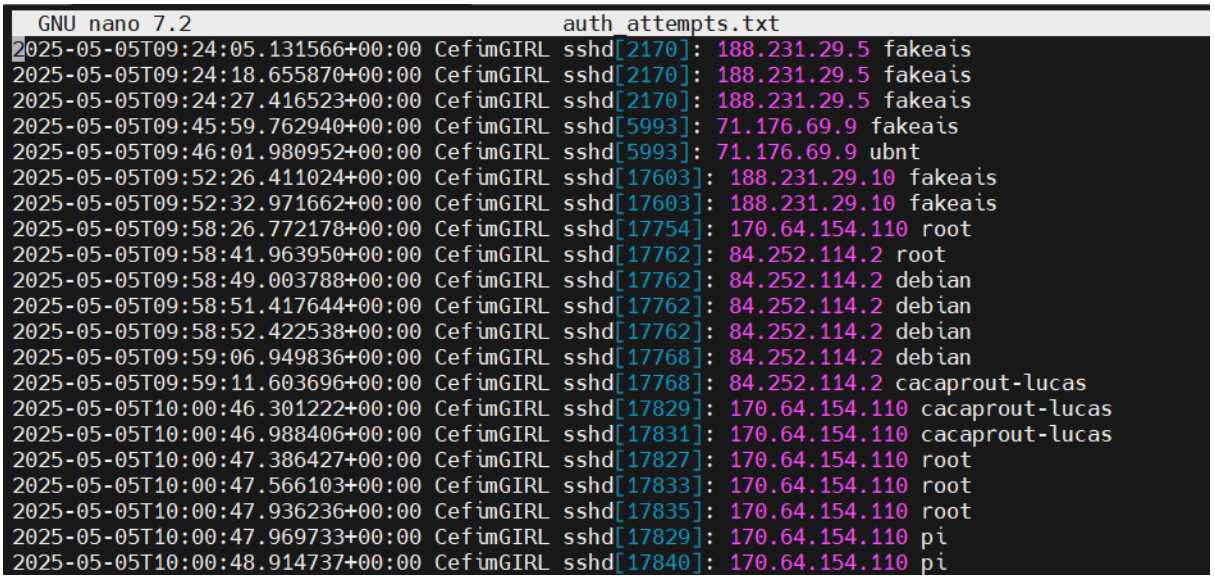
Crée un fichier avec les IP qui ont essayé de se connecter.

```
grep -E "Failed password|Invalid user" /var/log/auth.log   | grep -Eo '([0-9]{1,3
```

```
 GNU nano 7.2                                  ips_authlog.txt
104.28.216.246
170.64.154.110
188.231.29.10
188.231.29.5
20.82.114.17
71.176.69.9
79.127.134.11
84.252.114.2
```

fichier un peut plus complets:

```
grep -E "Failed password|Invalid user" /var/log/auth.log | awk '{
date=$1" "$2" "$3;
for (i=1; i<=NF; i++) {
if ($i=="from") ip=$(i+1);
if ($i=="invalid" && $(i+1)=="user") user=$(i+2);
if ($i=="for" && $(i+1)!="invalid") user=$(i+1);
}
print date, ip, user
}' > auth_attempts.txt
```



Whois - auth.log

```
grep -E "Failed password|Invalid user" /var/log/auth.log \
  | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' \
  | sort | uniq \
  | while read ip; do
    echo "=== $ip ==="
    whois $ip | grep -Ei '^(OrgName|organisation|netname|descr|country|OrgId
    echo ""
done
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash whois_auth_log.sh
=== 104.28.216.244 ===
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
OrgName:        Cloudflare, Inc.
OrgId:          CLOUD14
Country:        US

=== 104.28.216.246 ===
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
OrgName:        Cloudflare, Inc.
OrgId:          CLOUD14
Country:        US

=== 170.64.154.110 ===
CIDR:           170.64.128.0/17
NetName:        DIGITALOCEAN-170-64-128-0
OrgName:        DigitalOcean, LLC
OrgId:          DO-13
Country:        US

=== 188.231.29.10 ===
inetnum:        188.231.29.0 - 188.231.29.31
netname:        Linkt-MT-3943-FD
descr:          Linkt Customers
country:        FR

=== 188.231.29.5 ===
inetnum:        188.231.29.0 - 188.231.29.31
netname:        Linkt-MT-3943-FD
descr:          Linkt Customers
country:        FR

=== 20.82.114.17 ===
CIDR:           20.64.0.0/10, 20.48.0.0/12, 20.33.0.0/16, 20.128.0.0/16, 20.40.0.0/13, 20.34.0.0/15, 20.36.0.0/14
NetName:        MSFT
OrgName:        Microsoft Corporation
OrgId:          MSFT
Country:        US

=== 65.49.1.64 ===
CIDR:           65.49.0.0/17
NetName:        HURRICANE-9
OrgName:        Hurricane Electric LLC
OrgId:          HURC
Country:        US
CIDR:           65.49.1.0/24
NetName:        HURRICANE-CE2897-409C062A
OrgName:        The Shadowserver Foundation, Inc.
OrgId:          SF-1051
Country:        US

=== 71.176.69.9 ===
CIDR:           71.173.96.0/19, 71.180.0.0/16, 71.174.0.0/15, 71.173.128.0/17, 71.176.0.0/14
NetName:        VIS-BLOCK
OrgName:        Verizon Business
OrgId:          MCICS
Country:        US

=== 79.127.134.11 ===
inetnum:        79.127.134.0 - 79.127.134.255
netname:        CDNEXT-PAR
country:        FR
descr:          CDNEXT Paris

=== 84.252.114.2 ===
inetnum:        84.252.114.0 - 84.252.114.255
netname:        FR-ONETSOLUTIONS-NET-2-20181116
country:        AR
```

Whois - PCAP

Il faut installer tshark avant d’exécuter le script suivant :

```
tshark -r ssh-tcpdump.pcap -Y "tcp.dstport == 22" -T fields -e ip.src \
  | sort | uniq \
  | while read ip; do
    echo "=== $ip ==="
    whois $ip | grep -Ei '^(OrgName|organisation|netname|descr|country|OrgId
    echo ""
done
```

Voici la sortie du script :

```
fakeais@CefimGIRL:~$ sudo bash whois_pcap.sh
Running as user "root" and group "root". This could be dangerous.
=== 188.231.29.5 ===
inetnum:        188.231.29.0 - 188.231.29.31
netname:        Linkt-MT-3943-FD
descr:          Linkt Customers
country:        FR

=== 79.127.216.147 ===
inetnum:        79.127.216.0 - 79.127.216.255
netname:        CDN77-FRA
country:        DE
descr:          CDN77 Frankfurt
```

Maxmind GeoIP & GeoLite2

MaxMind GeoIP est une **base de données** (et une **bibliothèque logicielle**) permettant d'**associer une adresse IP à une localisation géographique** approximative : **pays**, **région**, **ville**, voire **fournisseur d'accès** ou **type de connexion**.

GeoLite2 est une **base de données gratuite** fournie par **MaxMind**, permettant d'effectuer une **géolocalisation approximative** des adresses IP publiques.

Elle est utilisée pour :

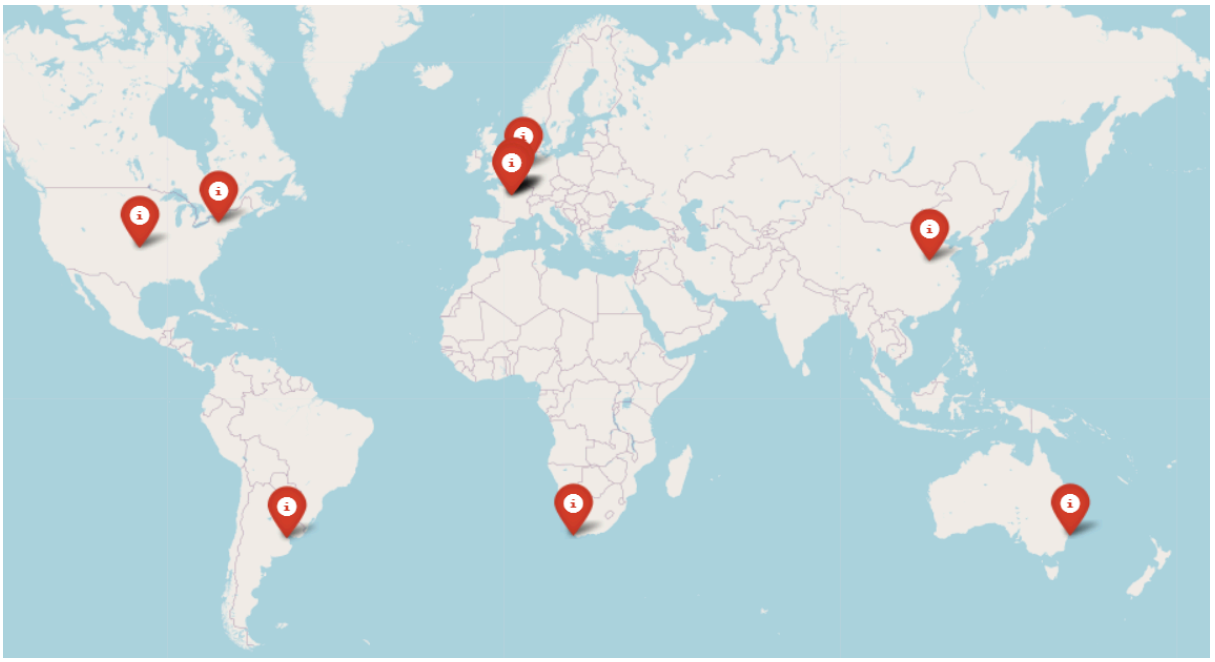
- Identifier les pays, la ville la région et parfois l'ASN
- Enrichir les logs réseau
- appliquer du filtrage géographique
- effectuer des analyse de comportement réseau

```
grep -E "Failed password|Invalid user" /var/log/auth.log \
| grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' \
| sort | uniq > ips.txt
```

```
source ~/geoip_env/bin/activate
```

```
python geoip-to-csv.py ips.txt
```

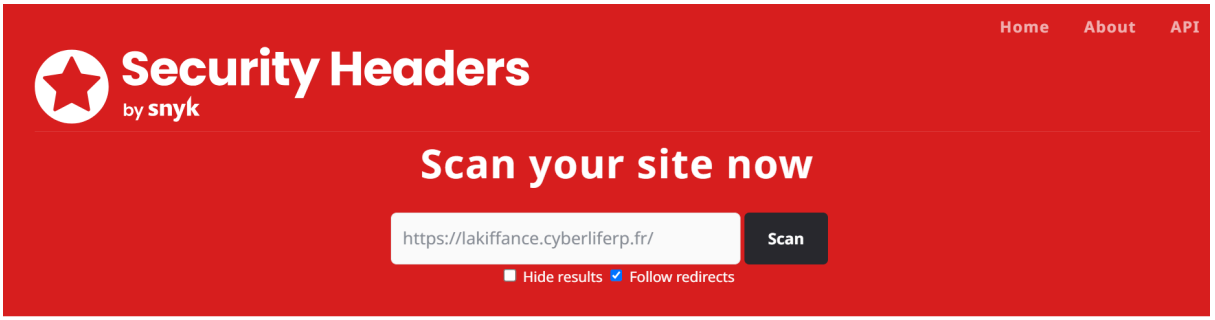
```
python3 csv-to-map.py geoip-results.csv
```

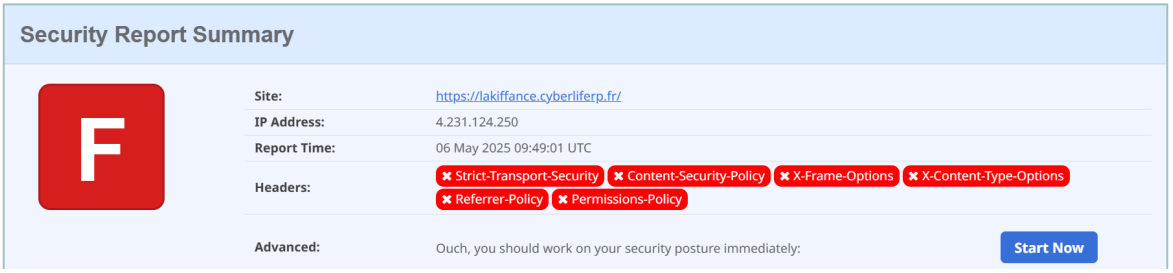


Security Headers

https://lakiffance.cyberliferp.fr/

Score avant hardening du site web :

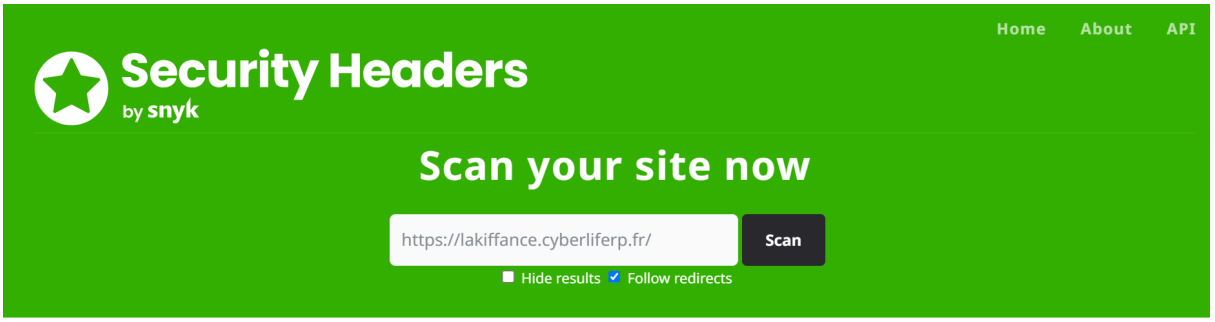


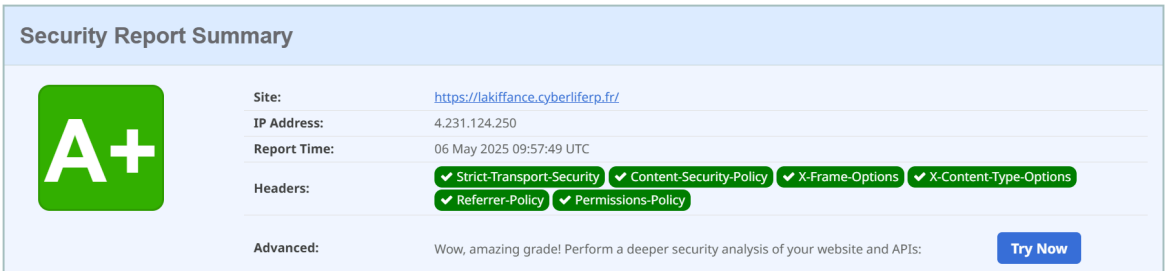


1. Ajouter dans le vhost apache du wordpress dans la partie ssl

```
    Header always set Strict-Transport-Security max-age=31536000

# Protection contre XSS et injections
Header always set Content-Security-Policy "default-src 'self'; script-src 'se

# Empêche l'inclusion du site dans une iframe
Header always set X-Frame-Options "SAMEORIGIN"

# Empêche le MIME sniffing
Header always set X-Content-Type-Options "nosniff"

# Politique de référent
Header always set Referrer-Policy "strict-origin-when-cross-origin"

# Désactivation des fonctionnalités non nécessaires
Header always set Permissions-Policy "geolocation=(), microphone=(), cam
```

Score final :



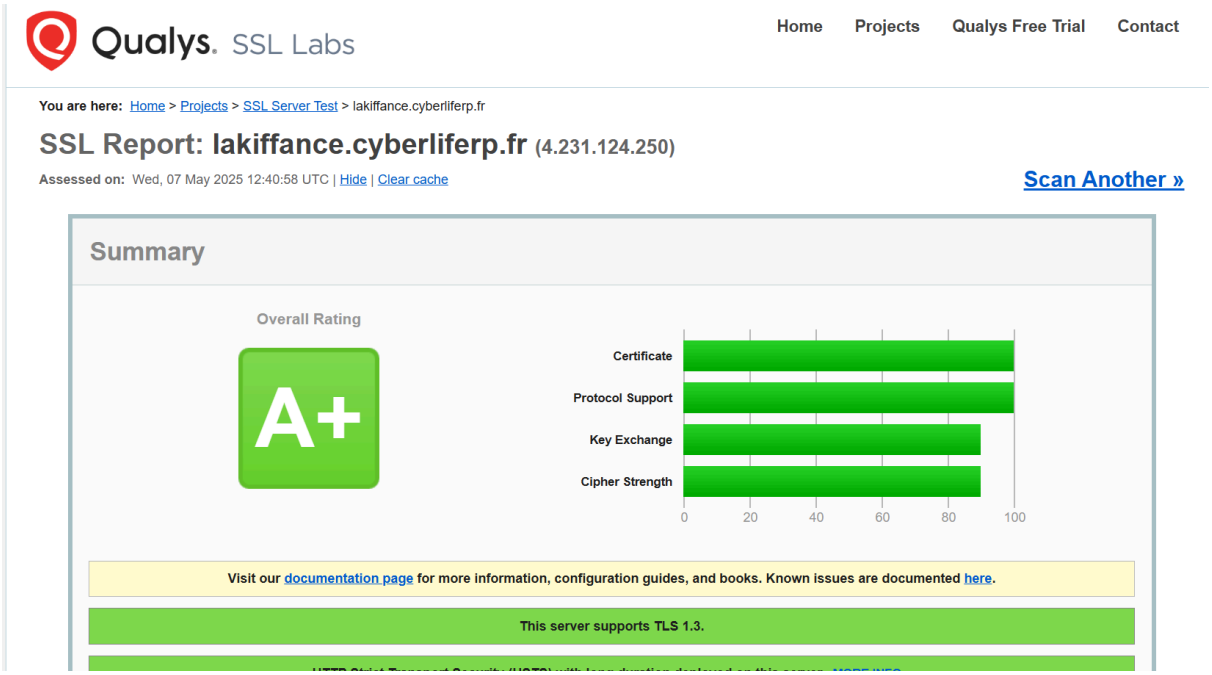


Voici notre Vhost :

```
<VirtualHost *:80>
    ServerName lakiffance.cyberliferp.fr
    ServerAlias www.lakiffance.cyberliferp.fr

    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/lakiffance_error.log
    CustomLog ${APACHE_LOG_DIR}/lakiffance_access.log combined
RewriteEngine on
RewriteCond %{SERVER_NAME} =www.lakiffance.cyberliferp.fr [OR]
RewriteCond %{SERVER_NAME} =lakiffance.cyberliferp.fr
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=perma
</VirtualHost>
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName lakiffance.cyberliferp.fr
    ServerAlias www.lakiffance.cyberliferp.fr
```

```
    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    Header always set Strict-Transport-Security max-age=31536000

    # Protection contre XSS et injections
    Header always set Content-Security-Policy "default-src 'self'; script-src 'se

    # Empêche l'inclusion du site dans une iframe
    Header always set X-Frame-Options "SAMEORIGIN"

    # Empêche le MIME sniffing
    Header always set X-Content-Type-Options "nosniff"

    # Politique de référent
    Header always set Referrer-Policy "strict-origin-when-cross-origin"

    # Désactivation des fonctionnalités non nécessaires
    Header always set Permissions-Policy "geolocation=(), microphone=(), cam

    ErrorLog ${APACHE_LOG_DIR}/lakiffance_error.log
    CustomLog ${APACHE_LOG_DIR}/lakiffance_access.log combined

  Include /etc/letsencrypt/options-ssl-apache.conf
  SSLCertificateFile /etc/letsencrypt/live/lakiffance.cyberliferp.fr/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/lakiffance.cyberliferp.fr/privkey.pe
  </VirtualHost>
  </IfModule>
```

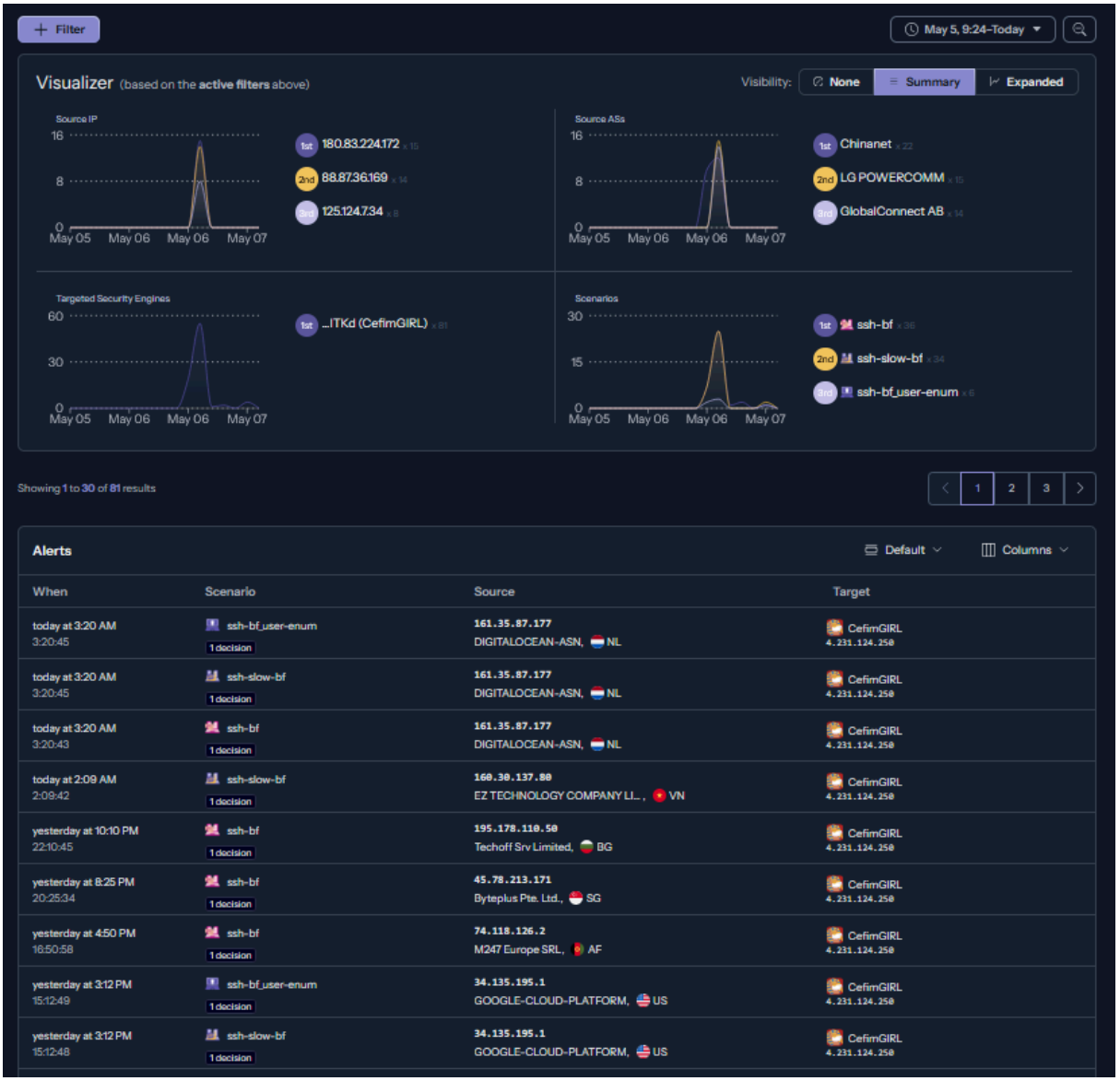
SSLLABS :



Crowdsec

```
sudo apt-get update
sudo apt-get install -y crowdsec
sudo cscli console enroll -e context "code_enrollment-crowdsec"
```

Alertes sur notre VM :



Hardening Wordpress avec Crowdsec

source

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-sshd   tcp  --  anywhere             anywhere             multiport dports ssh
CROWDSEC_CHAIN  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain CROWDSEC_CHAIN (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             match-set crowdsec-blacklists-1 src
DROP       all  --  anywhere             anywhere             match-set crowdsec-blacklists-0 src

Chain f2b-sshd (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

Lynis

Install

```
apt-get install lynis
```

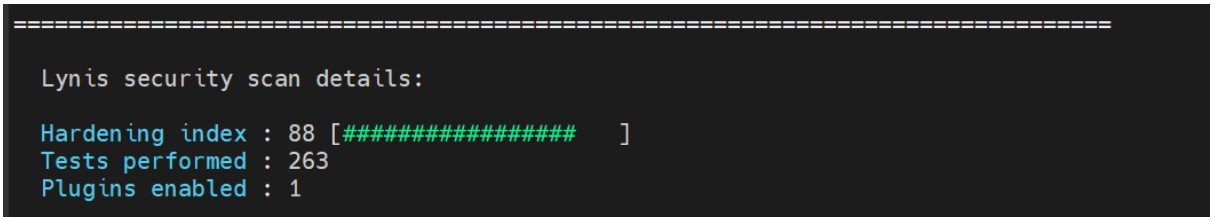
Lancer l'audit de sécurité

```
sudo lynis audit system
```

Options:

```
Alternative system audit modes
--forensics                : Perform forensics on a running or mounted syste
m
--pentest                  : Non-privileged, show points of interest for pentes
ting
```

Notre score Lynis Hardening Index : (de base 62)



Afin d'améliorer le score de Lynis :

Source list

Ajoutez les dépôts de sécurité au fichier `/etc/apt/sources.list` :

```
deb http://security.debian.org/debian-security bookworm-security main contr
```

```
sudo apt update
```

USB

Ajoutez une règle de **blacklist** pour empêcher le chargement du module **USB** :

```
echo "blacklist usb-storage" | sudo tee /etc/modprobe.d/disable-usb-storage
echo -e "blacklist firewire-core\nblacklist firewire-ohci\nblacklist sbp2" | sudo
sudo update-initramfs -u
```

ACCT

Le service **ACCT** enregistre chaque commande exécutée par les utilisateurs, avec des détails comme l'heure, la durée, l'utilisateur, etc. Pour l'install :

```
sudo apt install acct
sudo systemctl enable acct
sudo systemctl start acct
```

Sysstat

`sysstat` est un ensemble d'outils (comme `iostat`, `mpstat`, `sar`) pour collecter des statistiques sur les performances du système. Pour l'install :

```
sudo apt install sysstat
```

Pour a ctivez la collecte de données, modifiez `/etc/default/sysstat` et mettez :

```
ENABLED="true"
```

Redémarrez le service :

```
sudo systemctl enable sysstat
sudo systemctl restart sysstat
```

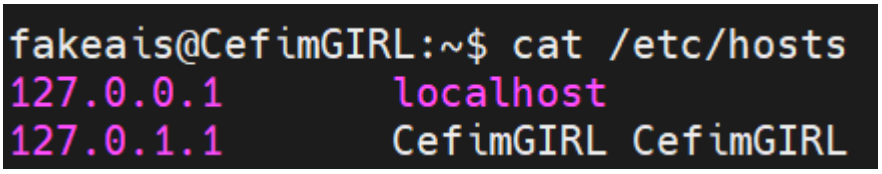
auditd

`auditd` peut enregistrer des actions critiques (accès à des fichiers sensibles, changements de privilèges, etc.). Pour l'install :

```
sudo apt install auditd audispd-plugins
sudo systemctl enable auditd
sudo systemctl start auditd
```

Hosts

Lynis recommande d'ajouter le **nom d'hôte complet (FQDN)** et l'adresse IP locale à `/etc/hosts`. Cela assure une résolution correcte du nom d'hôte, même sans DNS. Voici notre fichier :



debsums

`debsums` permet de **vérifier l'intégrité** des fichiers installés par des paquets `.deb` (en les comparant aux sommes MD5 d'origine). Utile pour détecter des fichiers modifiés ou compromis. Pour l'install :

```
sudo apt install debsums
```

Apparmor

```
fakeais@CefimGIRL:~$ sudo sudo apparmor_status
[sudo] password for fakeais:
apparmor module is loaded.
12 profiles are loaded.
12 profiles are in enforce mode.
   /usr/bin/man
   /usr/lib/NetworkManager/nm-dhcp-client.action
   /usr/lib/NetworkManager/nm-dhcp-helper
   /usr/lib/connman/scripts/dhclient-script
   /usr/sbin/chronyd
   /{,usr/}sbin/dhclient
   lsb_release
   man_filter
   man_groff
   nvidia_modprobe
   nvidia_modprobe//kmod
   tcpdump
0 profiles are in complain mode.
0 profiles are in kill mode.
0 profiles are in unconfined mode.
2 processes have profiles defined.
2 processes are in enforce mode.
   /usr/sbin/chronyd (575)
   /usr/sbin/chronyd (576)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
0 processes are in kill mode.
```

SYSCTL Configuration

```
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
kernel.kptr_restrict = 2
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv4.conf.all.log_martians = 1
fs.suid_dumpable = 0
```

IPTABLES

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-sshd   tcp  --  anywhere             anywhere             multiport dports ssh
CROWDSEC_CHAIN  all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:56789 ctstate NEW,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain CROWDSEC_CHAIN (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             match-set crowdsec-blacklists-1 src
DROP       all  --  anywhere             anywhere             match-set crowdsec-blacklists-0 src

Chain f2b-sshd (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

```
# Generated by iptables-save v1.8.9 (nf_tables) on Wed May  7 12:59:25 2025
*filter
:INPUT ACCEPT [12:1124]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [12:1548]
:CROWDSEC_CHAIN - [0:0]
-A INPUT -j CROWDSEC_CHAIN
-A CROWDSEC_CHAIN -m set --match-set crowdsec-blacklists-0 src -j DROP
COMMIT
# Completed on Wed May  7 12:59:25 2025
```

Questions SUDO

1. Quelles sont les différences entre
`sudo` et `su` ?

- `sudo` = faire une commande en mode "admin temporaire".
- `su` = se **transformer en root** (accès complet, session complète).

2. Pourquoi est-il préférable d'utiliser
`sudo` plutôt que se connecter directement en root ?
Parce que
`sudo` offre **plus de sécurité et de contrôle** :

- **Audit des actions** : toutes les commandes sont loguées (`/var/log/auth.log`).
- **Contrôle fin des permissions** : on peut accorder `sudo` pour **certaines commandes** seulement.
- **Réduction des erreurs** : on limite l'usage de privilèges élevés à des cas précis (moins de risques de casser le système).

- **Pas besoin de partager le mot de passe root** : chaque utilisateur a son propre accès.

3. Que se passe-t-il lorsqu'on lance `sudo` pour la première fois ?

- il affiche un **message d'introduction** (une fois) :

> "We trust you have received the usual lecture from the local System Administrator..."

- Puis il demande **le mot de passe de l'utilisateur courant**.
- Si l'utilisateur est dans le groupe `sudo`, la commande est exécutée avec les droits de root.

4. Quelles bonnes pratiques peut-on appliquer pour sécuriser l'usage de `sudo` ?

- **Exiger un mot de passe** (`NOPASSWD` désactivé sauf cas très particuliers).
- **Limiter la durée du cache de mot de passe** (`timestamp_timeout` dans sudoers).
- **Restreindre les commandes autorisées** dans le fichier `/etc/sudoers` (ex. : `user ALL=(ALL) /usr/bin/systemctl restart apache2`).
- **Éviter d'ajouter tout le monde dans le groupe `sudo`**.
- **Activer la journalisation et surveiller les logs**.
- **Tester les règles sudoers avec `visudo`** (évite les erreurs de syntaxe).

5. Qu'est-ce que la directive `NOPASSWD` dans le fichier `sudoers` ?

C'est une directive qui **désactive la demande de mot de passe** pour certaines commandes ou utilisateurs. Exemple :

```
user ALL=(ALL) NOPASSWD: /usr/bin/apt-get
```

➡️ Cela signifie que `user` peut faire `sudo apt-get` **sans mot de passe**.

6. Pourquoi sur Azure les `user` peuvent faire `sudo` sans mot de passe ?
Par défaut, Azure crée des utilisateurs avec une config `NOPASSWD`. Exemple typique dans `/etc/sudoers`

Il inclut par défaut @includir /etc/sudoers.d alors il faut le commenter.

Questions OPEN SSH

1. Quelle(s) version(s) du protocole SSH sont autorisées par défaut ? Est-ce sécuritaire ?

Par défaut, **seule la version SSHv2** est autorisée sur les systèmes modernes (depuis OpenSSH 7+).

- **SSHv2** est **considérée comme sécurisée** (chiffrement fort, algos modernes).
- **SSHv1** est **obsolète et vulnérable** (faille de MITM, chiffrement cassé).

Nous pouvons vérifier ce paramètre dans `/etc/ssh/sshd_config` :

```
Potocol 2
```

---

2. Quels sont les premiers paramètres de configuration à modifier pour durcir la sécurité d'un serveur OpenSSH ?

Voici une **liste de base** à appliquer dans `/etc/ssh/sshd_config` :

Paramètre	Recommandation
`PermitRootLogin no`	Interdire la connexion directe de root
`PasswordAuthentication no`	Forcer l'usage des **clés SSH**
`PermitEmptyPasswords no`	Interdire les comptes sans mot de passe
`ChallengeResponseAuthentication no`	Désactiver cette méthode ancienne
`MaxAuthTries 3`	Limiter les essais de connexion
`LoginGraceTime 30`	Temps max pour s'authentifier
`AllowUsers` ou `AllowGroups`	Limiter les accès à des comptes spécifiques
`Port 2222` *(optionnel)*	Changer le port par défaut pour éviter les scans automatisés

3. Pourquoi est-il recommandé de désactiver l'authentification par mot de passe et de privilégier l'authentification par clé ?

Avantages de l'authentification par clé :

- **Beaucoup plus difficile à forcer** qu'un mot de passe (clé privée 2048/4096 bits).
- Pas de transmission de mot de passe → **moins de risques d'interception**.
- Clé privée peut être protégée par une **passphrase**.
- Peut être combinée avec des outils de gestion centralisée (Ansible, Vault...).

Risques du mot de passe :

- Vulnérable au **brute-force**, à la **fuite** ou au **phishing**.
- Peut être réutilisé sur plusieurs systèmes — erreur humaine fréquente.

---

4. Comment restreindre les utilisateurs autorisés à se connecter via SSH ?

Dans `/etc/ssh/sshd_config` :

```
AllowUsers alice bob
```

ou

```
AllowGroups sshusers
```

Pour un contrôle plus propre, il est préférable de **créer un groupe (`sshusers`)** et de n'y ajouter que les utilisateurs autorisés.

---

5. Un service SSH doit-il être exposé directement sur Internet ?

Non, sauf si **absolument nécessaire**.

Conditions strictes si exposé :

- Authentification par **clé uniquement**
- Filtrage IP via un **pare-feu** ou un **bastion**
- Journalisation et alertes actives (fail2ban, auditd...)
- Port non standard (optionnel, pour réduire les scans automatisés)

Alternatives recommandées :

- **VPN** (ex : WireGuard, OpenVPN)
- **Bastion SSH (jump host)** : un seul point d'entrée sécurisé
- **Port knocking** ou authentification multi-facteur

---

6. Quelle serait votre procédure d'audit rapide pour vérifier si un serveur SSH est correctement sécurisé ?

```
# 1. Vérifier les ports ouverts
ss -tuln | grep ssh

# 2. Revue du fichier de config
grep -Ei 'PermitRootLogin|PasswordAuthentication|Protocol|Port|AllowUsers' /etc/ssh/sshd_config

# 3. Vérifier les logs SSH
tail -n 100 /var/log/auth.log

# 4. Vérifier la version du serveur
ssh -V

# 5. Scanner à distance avec nmap
nmap -sV -p 22 --script ssh2-enum-algos <IP>

# 6. Lister les utilisateurs avec un accès SSH
grep '/bin/bash' /etc/passwd

# 7. Vérifier les clés autorisées
cat /home/*/.ssh/authorized_keys
```

---

7. Dans un contexte de PSSI (Politique de Sécurité des SI), quelles règles devraient encadrer l'usage de SSH par les administrateurs ?

Les règles PSSI typiques pour SSH :

- **Clé SSH obligatoire**, protégée par **passphrase**
- Accès **limité à certains comptes** (via `AllowGroups`)
- **Rotation périodique des clés**

- **Journalisation obligatoire** de toutes les connexions
- **Authentification MFA** si possible (via `pam_u2f`, TOTP...)
- Interdiction de **connexion root directe**
- Accès SSH via un **bastion** ou VPN
- Formation régulière sur la gestion des clés et pratiques sûres
- Audits réguliers et vérifications des accès

Voici un exemple d'un fichier sshd sécurisé :

```
`Port 2222                    # Change le port (optionnel, mais utile contre les sc
Protocol 2                    # Utiliser uniquement SSH v2
AddressFamily inet            # IPv4 uniquement (si IPv6 inutile)
ListenAddress 0.0.0.0         # Ou adresse IP privée spécifique si serveur loc

# `----------------------`

# `Authentification`

# `----------------------`

`PermitRootLogin no           # Interdit l'accès direct en root
PasswordAuthentication no     # Force l'usage de clés SSH
PermitEmptyPasswords no       # Interdit les comptes sans mot de passe
ChallengeResponseAuthentication no
UsePAM yes                    # Autorise l'intégration PAM (utile pour MFA)
AuthenticationMethods publickey   # Exige les clés publiques`

# `----------------------`

# `Connexions et sécurité`

# `----------------------`

`LoginGraceTime 30            # Temps max pour s'authentifier
MaxAuthTries 3                # Limite les essais de login
MaxSessions 2                 # Limite les sessions SSH par utilisateur
ClientAliveInterval 300       # Envoie un ping toutes les 5 min
```

```
ClientAliveCountMax 2           # Déconnecte après 10 min d'inactivité`

# `----------------------`

# `Utilisateurs autorisés`

# `----------------------`

`AllowGroups sshusers           # Seuls les membres du groupe sshusers peuv

# `----------------------`

# `Logs & audit`

# `----------------------`

`LogLevel VERBOSE               # Journalise les clés utilisées
SyslogFacility AUTH`

# `----------------------`

# `Clés acceptées`

# `----------------------`

`HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_rsa_key
PubkeyAcceptedKeyTypes ssh-ed25519,ssh-rsa`

# `----------------------`

# `Restrictions supplémentaires`

# `----------------------`

`X11Forwarding no
AllowTcpForwarding no
```

```
PermitTunnel no
Banner /etc/issue.net          # Message d'avertissement de sécurité (optionn
```

Pour se connecter en SSH via clé :

Questions APACHE2

1. Que fait une installation par défaut d'Apache ?

Une installation par défaut d'Apache (ex. via `apt install apache2`) :

- Installe le binaire du serveur HTTP (`httpd`)
- Crée un site par défaut (souvent dans `/var/www/html`)
- Active un **VirtualHost par défaut** sur le port 80
- Démarre le service `apache2` automatiquement
- Crée une arborescence classique :
  - `/etc/apache2/` → fichiers de config
  - `/var/www/html` → répertoire du site par défaut
  - `/var/log/apache2/` → logs `access.log` et `error.log`

---

2. Quels sont les risques d'un serveur Apache "sortie d'usine" ?

- **Site par défaut accessible** → donne des infos sur la version Apache
- **Indexation des répertoires activée** (si pas désactivée)
- **Pages d'erreur standards** → pas de masquage de l'architecture
- **Modules inutiles activés** (ex : `status`, `autoindex`, `userdir`, etc.)
- **Bannières d'information actives** (`ServerSignature`, `ServerTokens`)
- **Pas de chiffrement HTTPS configuré**
- **Fichiers sensibles mal protégés** si mal configuré (`.git`, `.env`, etc.)

---

3. Pourquoi isoler les applications Apache dans des VirtualHosts bien définis ?

Isoler les applis web via des `VirtualHosts` permet de :

- **Séparer les permissions** (fichiers, logs, modules)
- **Journaliser indépendamment** chaque site
- **Limiter l'impact d'une faille** à un seul site
- Faciliter la **gestion des certificats SSL/TLS**
- Faciliter l'**application de politiques spécifiques** (CSP, headers, limites de taille, etc.)

---

4. Quelles traces doit-on surveiller pour détecter une compromission Apache ?

Surveille ces fichiers/logs :

Fichier ou ressource	Ce qu'il faut chercher
`/var/log/apache2/access.log`	Requêtes anormales, payloads suspects (injections, uploads, shell...)
`/var/log/apache2/error.log`	Erreurs 500 fréquentes, appels à des fichiers inconnus
`/var/log/auth.log`	Connexions SSH anormales (si Apache est sur un serveur unique)
Fichiers modifiés récemment	`find /var/www -mtime -1` pour voir les modifs récentes
Requêtes POST vers fichiers `.php`, `.txt`, `.jpg`	Tentatives d'exécution de WebShell
Headers HTTP suspects	User-Agent vides, curl, python-requests, etc.

---

5. Comment détecter un WebShell ou une backdoor dans un serveur Apache ?

Quelques techniques efficaces :

- **Scanner les fichiers PHP** :

```
grep -Ri "eval(" /var/www
grep -Ri "base64_decode(" /var/www
grep -Ri "system(" /var/www
```

- **Chercher des fichiers suspects** :

- noms obscurs : `cmd.php`, `r57.php`, `shell.php`, `up.php`, `1.php`
- extensions déguisées : `.jpg.php`, `.txt.php`

- **Analyser les accès** :

```
grep "\.php" /var/log/apache2/access.log | grep POST
```

- **Mettre en place un HIDS (ex. : `AIDE`, `OSSEC`, `Tripwire`)** pour surveiller les changements de fichiers
- **Utiliser un scanner comme** :
  - `ClamAV` (avec base signatures PHP)
  - `Lynis`, `rkhunter` (détection de rootkits)
  - `Chkrootkit`
  - Outils dédiés à l'analyse de WebShells (comme LMD – Linux Malware Detect)

---

6. En plus des éléments ci-dessus, quelles ressources utiliser ?

Voici quelques **ressources utiles** supplémentaires :

Outils techniques :

- **ModSecurity** (WAF pour Apache)
- **Fail2ban** (protection contre le brute-force)
- **Apache Security Headers** :
  - `X-Content-Type-Options: nosniff`
  - `X-Frame-Options: DENY`
  - `Content-Security-Policy: default-src 'self'`
  - `Strict-Transport-Security`
- **Audit avec `ssllabs.com`** pour la sécurité HTTPS
- **Nikto / Nmap** pour scanner les vulnérabilités connues

Guides & recommandations :

- CIS Benchmark Apache HTTP Server
- OWASP Secure Configuration Guidelines

- Guides ANSSI (ex : RGS)

Hardening des configurations d'Apache2

- **Hardening des en-têtes HTTP de sécurité**
  - Testez votre configuration avec **Security Headers**

```
/etc/apache2/conf-available/security.conf
```

- **Hardening des paramètres SSL/TLS pour Apache2**
  - Testez votre configuration SSL avec **sslabs**

```
/etc/letsencrypt/options-ssl-apache.conf
```

- **Pare-feu applicatif (WAF)**
  - **Mod Security 2 & OWASP CRS**
  - **GitHub OWASP CRS**