Debian

SSHD

- Changement du port par défaut
- Authentification par clé publique
- Désactivation de la connexion en tant que root

LXC

- Conteneur lançable uniquement en root

Apache (*:8000)

- ExeCGI bloqué dans tous les dossiers (empêche d'interpréter du python, perl, bahs etc)
- Activation du chiffrement TLS via certificat auto signé (on a viré l'utilisation de TLS 1.1 et de TLS)
- Redirection des erreurs sur le site wordpress
- Désactivation de l'indexation des directory, des signatures apache, du TraceEnable et de la version de php dans le header
- Désactivation des erreurs php (pour éviter d'avoir une fuite de code affiché dans l'erreur), des fonctions sensibles type shell_exec ou system qui permettent de run des commandes shell en php
- Login apache devant le /cacti et le /munin
- Logging des erreurs

Munin

- Serveur qui monitore localhost et le conteneur

Cacti

- Serveur qui monitore localhost & le conteneur (SNMPv2 pour le moment, le v3 faisait buguer Cacti)

MariaDB

- écoute en localhost
- secure_installation run pour virer les tables/anonyme user etc + changement mdp root@localhost

Crontab

- Mâj de sécu des packages via cron-apt

Iptables

- On rejette les scan nmap de type fragment, null, xmas et toutes les connexions de type "NEW" qui ne commencent pas par un SYN
- on autorise la machien debian à communiquer avec le conteneur sur les protocoles DHCp/DNS
- on allow le traffic sur localhost
- on accepte toutes les connexions qui sortent de la debian
- on autorise le traffic sortant du conteneur
- on accepte le traffic INPUT et FORWARD venant de l'interface côté publique du serveur debian (ens33) sur le port 80 (http conteneur), 443 (https conteneur), 2121 (ftp conteneur)
- on accepte le traffic INPUT venant de l'interface côté publique du serveur debian (ens33) sur le port 2202 (ssh debian), 8000 (apache debian en https)
- on log les INPUT/OUTPUT/FORWARD Denied
- on reject tous les INPUT/FORWARD qui ont pas match les règles précédentes
- On route les paquets qui ont pour dest le port 80, 443, 2121 vers le conteneur
- On transforme l'ip les paquets venant du sous réseau 10.0.10.0/24 donc privé (S.R du conteneur) en ip "publique"

Hardening

- Machine chiffrée et partitions séparées (fait à l'installation)
- fichier .htpasswd qui contient le user/hash mdp du Login apache accessible qu'en lecture par le user www-data
- Désactivation de l'historique des users, redirection de l'historique des commandes mysql vers /dev/null
- User normal avec mdp fort + sudo

Conteneur LXC ALPINE

Alpine car distribution très légère basée sur busybox, pratiquement aucun service d'installé de base (image de 3Mo) donc surface d'atatque très faible.

Apache (*:80, *:443)

- Même chose que plus haut sauf qu'on redirige tout le :80 sur le :443 (tls)
- On réécrit les URL wordpress pour que ce soit plus beau
- On vire l'accès à l'API xmlrpc.php du wordpress (prévention bruteforce/pingback)
- On créer une règle anti DDOS pour avoir un nombre de session limitée

MariaDB

- Même chose que sur la debian

phpMyAdmin

- Protégé par un la mire de login apache
- réécriture du lien, c'est pas /phpmyadmin mais /adminmyphp (pour montrer qu'on sait faire)
- Désactivation de la possibilité de se connecter en tant que root
- Injection d'une chaine aléatoire de 32 caract-res dans le blowfish_secret(pour le chiffrement des cookies de sessions je crois)

VSFTPD

- On vire la connexion en user anonyme
- On autorise à se co en utilisant les users locals à la machone
- umask par défaut = 022 (les fichiers/dossiers sont créés avec les droits 755)
- On bloque les users dans leur home directoru `chroot_local_user`
- Port d'écoute à 2121
- Activation de TLS 1.2 pour avoir des flux chiffrés

Wordpress

- creation d'une DDB + user dédié

- installation de 2FA (google auth) + recaptacha v3 + Firewall/IPS interne à wordpress pour update les plugins plus à jour, blacklist les IP qui font des conneries etc (Wordfence)

Munin & snmp

- munin-node pour joindre le serveur munin sur le debian pour monitorer
- snmp pour communiquer avec le serveur snmp de cacti