Documentation Labtainers

Ce document explique comment installer, configurer et utiliser Labtainers pour réaliser des laboratoires pratiques en cybersécurité.

1. Présentation

Labtainers est une plateforme open-source fournissant des environnements virtuels Docker préconfigurés. Elle permet de déployer rapidement des laboratoires de cybersécurité pour l'apprentissage et l'entraînement, sans se soucier de la configuration d'infrastructures.

2. Installation

2.1. Sous Linux (Ubuntu/Debian)

```
sudo apt update  
sudo apt install docker.io docker-compose python3-pip git  
pip3 install --user labtainer
```

2.2. Clonage du dépôt Labtainers

```
git clone https://github.com/mfthomps/Labtainers.git  
cd Labtainers
```

3. Configuration

Ajouter l'utilisateur au groupe Docker :

```
sudo usermod -aG docker $USER
```

Recharger la session :

```
newgrp docker
```

4. Utilisation de base

4.1. Lister les laboratoires disponibles

```
labtainer list
```

4.2. Démarrer un laboratoire

```
labtainer start <nom_du_lab>
```

Exemple : `labtainer start lab00`

4.3. Arrêter et nettoyer

```
labtainer stop all # arrêter tous les conteneurs
```

```
labtainer clean # supprimer les ressources créées
```

4.4. Aide

```
labtainer --help
```

5. Options avancées

- Fork : `labtainer --fork=N` pour lancer plusieurs labos en parallèle.

- Timeout : `--max-run-time=<secondes>` pour limiter la durée d'exécution.
- Configuration : modifier `labtainer.config` dans chaque labo pour ajuster nom, description, niveaux.

6. Création et personnalisation de laboratoires

Structure recommandée d'un labo :

`mon_labo/`

├─ `setup.sh` # script d'initialisation

├─ `dockerfiles/` # définitions Docker

├─ `scripts/` # outils et exercices

└─ `labtainer.config` # métadonnées (nom, description, prérequis)

1. Copier un modèle depuis `templates/`.
2. Adapter les Dockerfiles et `setup.sh`.
3. Ajouter/modifier les scripts dans `scripts/`.
4. Configurer `labtainer.config`.

7. Dépannage

- Vérifier Docker : `docker ps`
- Consulter les logs : `docker logs <container_id>`
- Suppression forcée : `docker rm -f <container_id>`

8. Bonnes pratiques

- Mettre à jour régulièrement les images Docker (docker pull).
- Versionner vos laboratoires avec Git.
- Isoler les environnements réseau (ex. VLAN dédié).