UNIVERSITÉ DE LORRAINE | IUT nancy Charlemagne

BUT informatique 3A, parcours DACS

(pré-)SAÉ de projet tuteuré

Projet audit et déploiement
20 octobre 2025

1 Objectif

Dans ce projet, il s'agit de développer une solution d'audit de systèmes Linux, afin de refaire le point sur les commandes de base, notamment celles permettant d'obtenir des informations sur de tels systèmes. Le projet se déroule en trois étapes progressives :

1. manipulation et extraction d'informations systèmes via les commandes Unix,
2. automatisation et enrichissement par des scripts Ruby,
3. conteneurisation de l'environnement et déploiement *via* Docker.

Le travail s'effectue par groupe de 3 étudiants. Ce sujet est également un prétexte pour parfaire vos notions sur les différentes commandes et technologies à utiliser. Il est vivement conseillé de profiter de cette occasion pour « creuser » les concepts utilisés. L'environnement Linux est **obligatoire** pour les travaux demandés.

L'évaluation finale se composera du rendu des ressources demandées ainsi qu'une mini-soutenance (jeudi en fin de matinée) pour répondre du travail réalisé et de la culture générale technique sous-jacente.

2 Descriptif des étapes à réaliser

1 Audit système

Objectif : savoir localiser, collecter et formater des informations système pertinentes à l'aide des commandes Unix usuelles.

Dans cette partie, il s'agit d'être capable de retrouver les informations système suivantes au moment où elles sont récupérées :

- nom de la machine, distribution, version du noyau,
- *uptime*, charge moyenne, mémoire et swap disponibles et utilisés,
- liste des interfaces réseau : adresses MAC et IP associées,
- liste des utilisateurs humain (`uid` $\geqslant 1000$) existants, en distinguant ceux actuellement connectés,
- espace disque par partition (disponible, utilisé),
- processus les plus consommateurs de CPU et de mémoire (paramétrer par un seuil),
- processus les plus consommateurs de trafic réseau (paramétrer par un seuil),
- présence et statut (*up*, *down*) de certains services clés (ex. : `sshd`, `cron`, `docker`).

Commentaires sur les travaux demandés :

- les commandes utilisées doivent être standards (`uname`, `lsb_release`, `free`, `df`, `ps`, `systemctl`, *etc.*),

— pour chaque information demandée, il faut prévoir un descriptif, des commentaires, sur l'approche observée et l'usage réalisé des commandes (quels paramètres ? sur quelle ressource ? *etc.*) Inutile d'être trop verbeux sur les informations facilement récupérables. Certaines d'entre elles peuvent être abordées *via* différents points de vue : justifier l'approche que vous avez finalement suivie.

❷ Automatisation *via* un script Ruby

Objectif : agréger la récupération de toutes les informations précédentes *via* un script, gérer le formatage du résultat.

Dans cette partie, il s'agit de réaliser un script Ruby *paramétrable*, permettant l'automatisation de la récupération de toutes les informations demandées (page 1). Ce script devra supporter différents paramètres (aucun dialogue interactif n'est autorisé), permettant :

— de récupérer lors de son exécution toutes les informations évoquées dans la première partie,
— au choix d'afficher les informations récupérées (avec un formatage ergonomique) ou l'enregistrement de ces informations dans un fichier au format JSON.

Commentaires sur les travaux demandés :

— le script doit être versionné en utilisant `git` : l'adresse du dépôt correspondant (public au moment du rendu final) devra être communiquée lors du rendu,
— le code doit être propre et commenté.

❸ Déploiement dans un environnement Docker

Objectif : isoler et déployer l'outil d'audit dans un container Docker.

Dans cette partie, l'objectif est de mettre à disposition l'outil réalisé par le biais d'un container Docker. Dans ce cadre, le travail demandé consiste à :

1. créer un `Dockerfile` permettant :
   — d'installer Ruby et les dépendances nécessaires,
   — de copier le script développé,
   — de définir un point d'entrée (`ENTRYPOINT`) exécutant le script réalisé
2. vérifier que le conteneur créé peut récupérer les informations de ses pairs via SSH (clé publique/privée).

3 Résultats attendus

— Un dépôt Git avec le code Ruby pour l'audit.
— Un mini-rapport avec les explications de la première partie et les fichiers de configuration Docker.
— Une démonstration du résultat obtenu jeudi 23 octobre en fin de matinée.