Université Bretagne Sud
IUT de Vannes
Département Informatique

BUT 3 – R5.A.04.Qualité algorithmique
TP 3 : analyse d'artefacts logiciels pour évaluer la sécurité

Nous allons utiliser dans ce TP l'outil « `docker scout` » pour analyser des images Docker d'un logiciel. Cet outil liste les éléments de la nomenclature du logiciel (*SBOM : Software Bill of Material*) : les bibliothèques, frameworks, … les versions utilisées, ... Ensuite il consulte plusieurs base de données (*Advisory Databases*) pour identifier les vulnérabilités CVE. Ces bases de données sont mises à jour régulièrement.

Les vulnérabilités signalées par ce genre d'outils sont assez variées. Par exemple, il notifie :

1. s'il y a des dépendances obsolètes ou vulnérables : utilisation d'anciennes versions de bibliothèques ou d'outils présentant des vulnérabilités connues.
2. s'il y a des privilèges inutiles : exécution du conteneur avec des privilèges root.
3. s'il y a des paquets inutiles : les paquets qui ne sont pas nécessaires à l'application.
4. s'il y a des secrets codés en dur : y compris les jetons ou les informations d'identification dans l'image.
5. s'il y a des dépendances non « épinglées » : installation de paquets sans spécification de versions exactes.
6. s'il n'y a aucune spécification d'utilisateur ou de groupe : le fait de ne pas spécifier d'utilisateur autre que l'utilisateur root pour exécuter l'application.

Télécharger l'archive ZIP donnée dans la page Moodle du cours.

Décompresser l'archive.

Construire l'image Docker vulnérable avec le premier fichier Dockerfile : 1.Dockerfile:

```
docker build -t vulnerable-image -f 1.Dockerfile .
```

Lancer docker scout sur l'image précédente (`docker login` d'abord sur docker hub) :

```
docker scout cves vulnerable-image
```

Corriger les vulnérabilités signalées quand c'est possible et reconstruire une nouvelle image.

Explorer les différentes commandes et options de `docker scout` pour filtrer le type de vulnérabilités à afficher, obtenir des recommandations, enregistrer un dépôt d'images pour une analyse continue (à chaque push – dans ce TP, on l'utilise pour des tâches ponctuelles via la CLI), … La documentation complète de l'outil est fournie ici : https://docs.docker.com/scout/

Dans les résultats d'analyse, docker scout affiche le niveau de sévérité de certaines vulnérabilités mais aussi les valeurs d'une métrique connue sous le nom CVSS. Comment est calculée cette métrique ? Que veut dire le vecteur suivant ?

```
CVSS Vector : CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
```

Lire la page ci-dessous pour mieux comprendre :
https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System

Dans votre rendu, expliquer l'un des vecteurs obtenus lors de votre analyse dans un fichier texte.

D'autres Dockerfiles pour construire des images vulnérables sont fournis sur Moodle.
Identifier les nouveaux types de vulnérabilités (non mentionnées dans les autres) sur chacune des images. Essayer de le faire sans docker scout juste en inspectant les Dockerfiles, puis lancer docker scout pour vérifier. Le faire sur deux Dockefiles minimum.

Rendre sur Moodle des Dockerfiles annotées avec les vulnérabilités et la façon de les corriger quand c'est possible.

Analyser avec `docker scout` quelques images Docker créées, avec des Dockerfile à vous notamment, ou simplement utilisées dans des orchestrations (notamment celles des serveurs de base de données) dans vos projets. Corriger les éventuelles vulnérabilités signalées.

À la fin du TP, ne pas oublier de supprimer sur votre machine les images Docker inutiles :

```
docker image ls
docker rmi <image_id>
```