Rapport de SF - DEVOPS Partie 3

La troisième session de formation a été centrée sur la présentation des pipelines CI/CD par les participants et sur les exigences de sécurisation des infrastructures et des processus de build, ainsi que sur les différentes analyses de sécurité.

Cette session de formation a commencée par une présentation des exigences de sécurisation pour différents niveaux de sécurité :

Présentation des Exigences de Sécurisation

1. Exigences de Sécurisation

- **Configuration de l'Infrastructure**
    - **Renforcement des serveurs** : Les serveurs doivent être correctement renforcés (hardening).
    - **Mise à jour régulière des composants** : Les composants du serveur doivent être mis à jour régulièrement pour éviter les failles de sécurité.
    - **Séparation des éléments d'infrastructure** : Chaque élément doit être installé dans des zones sécurisées et séparées des autres applications.
    - **Scannage régulier des actifs** : Les actifs doivent être régulièrement scannés pour détecter les failles de sécurité.
- **Processus de Build (DevSecOps)**
    - **Composants non-supportés** : L'application ne doit pas contenir de composants non-supportés par un éditeur.
    - **Avertissement des vulnérabilités** : Les outils de build doivent alerter l'équipe sur les composants vulnérables ou non-sécurisés.
    - **Automatisation et sécurisation du build** : Le processus de build doit être automatisé et sécurisé, incluant des tests de sécurité.
    - **Vérification de la sécurité du code** : Analyse du code et des librairies pour vérifier la sécurité avant le déploiement.
- **Dépendances**
    - **Effacement des informations sensibles** : Les informations utilisées lors du développement doivent être effacées de la version mise en production.

    - **Validation des composants externes** : Les composants externes doivent être validés et provenir de dépôts de confiance.

- **Sécurisation des Headers HTTP**

    - **Content-Type** : Les réponses HTTP doivent inclure un Content-Type approprié.
    - **Protection contre XSS et injections JavaScript** : Toutes les mesures nécessaires doivent être prises pour bloquer les XSS et les injections JavaScript.
    - **HTTP Strict Transport Security (HSTS)** : Les réponses doivent inclure les headers HSTS pour sécuriser les communications.

2. Analyses de Sécurité

- **Analyses Statiques et Dynamiques**
    - **SAST (Static Application Security Testing)** : Analyse statique du code source pour détecter les vulnérabilités avant l'exécution.
    - **DAST (Dynamic Application Security Testing)** : Analyse dynamique des applications en cours d'exécution pour identifier les failles.
    - **YAST (Yet Another Security Testing)** : Autres méthodes d'analyse de sécurité pour assurer une couverture complète.

Voici un extrait des exigences de sécurisation des infrastructures et des processus de build présentées lors de la session de formation. Ces exigences sont essentielles pour garantir la sécurité des applications et des systèmes développés :

3. Présentation des Pipelines CI/CD

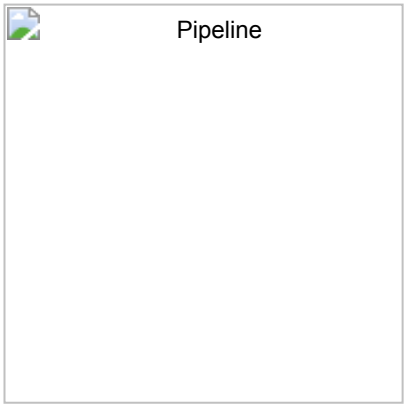
Par la deuxième partie de cette session, les participants ont présenté les pipelines CI/CD qu'ils ont mis en place pour leurs projets. Ces présentations ont permis de discuter des bonnes pratiques, des défis rencontrés et des solutions apportées pour améliorer la qualité et la sécurité des processus de build et de déploiement.

Il y avait donc la présentation de la pipeline CI/CD de chaque équipe, suivie d'une discussion sur les points forts et les améliorations possibles. Les points suivants on été relevés :

- Il est essentiels de séparer les environnements de développement, de test et de production pour garantir la qualité et la sécurité des déploiements.

- Les outils d'analyse de sécurité doivent être intégrés dans le processus de build pour détecter les vulnérabilités avant le déploiement.
- Les tests automatisés doivent être exécutés à chaque étape du pipeline pour garantir la qualité et la fiabilité des applications.
- Les processus de build et de déploiement doivent être documentés et automatisés pour assurer la reproductibilité et la cohérence des déploiements.
- Utiliser un système pour les variables d'environnement et les secrets pour garantir la sécurité des informations sensibles.
- La branche dev est la branche d'intégration continue, elle a donc son propre pipeline CI.

Voici un exemple complet d'une pipeline CI/CD présenté lors de la session de formation :



Ces deux sessions de formation ont permis aux participants d'acquérir une compréhension approfondie des pratiques essentielles en gestion de projet, en développement logiciel et en sécurisation des processus de build et des infrastructures. Les discussions et les démonstrations ont offert des perspectives pratiques et concrètes pour améliorer la qualité et la sécurité des projets.