Application : Versionner et améliorer sa solution en continu

Table des matières

I. Application : Versionner et améliorer sa solution en continu

Conseil

Cette application permet de valider vos connaissances et acquisitions de la compétence, en lien avec le référentiel du parcours visé. Elle ne participe pas à l'obtention de votre parcours, mais vous permet de vous entraîner à l'appropriation de la pratique professionnelle attendue.

Veillez à la qualité de votre syntaxe.

Durée indicative : 1 heure(s)

Documents autorisés : tous les documents sont autorisés

Matériel autorisé : tous les matériels sont autorisés

II. Exercice : Partie 1 : Gestion des versions

Vous travaillez dans une entreprise d'édition de logiciel. Le versionning est géré manuellement. C'est-à-dire à chaque fois une nouvelle version est développée, le code source est enregistré sous forme de .zip. Cette méthode n'est pas pratique, car il est difficile de se retrouver dans les différents fichiers, de plus elle rend le travail en équipe presque impossible. Votre responsable vous demande de trouver une solution.

Question 1

[solution n°1 p.5]

1.1. Quelle solution proposez-vous ?

Question 2

[solution n°2 p.5]

1.2. Qu'est-ce que GitLab ?

Question 3

[solution n°3 p.5]

1.3. Quelles sont les étapes à suivre afin de le mettre en place un dépôt GitLab ?

III. Exercice : Partie 2 : Pentest

Vous travaillez dans une agence de création de sites web. Suite au déploiement du nouveau site, votre chef vous demande de faire un Pentest afin de détecter les vulnérabilités.

Question 1

[solution n°4 p.6]

2.1. Qu'est-ce qu'un Pentest ?

Question 2

[solution n°5 p.6]

2.2. Présentez les différents types de tests.

Question 3

[solution n°6 p.6]

2.3. Quelles sont les étapes d'un Pentest ?

Solutions des exercices

p. 3 Solution n°1

Afin de remédier à ses problèmes, il faut utiliser un outil de versionning.

Les outils de gestion des versions sont utilisés pour modifier des fichiers, voire des documents. Ils enregistrent toutes les versions créées dans une archive, avec un horodatage et un identifiant unique, de manière à ce que les anciennes données puissent être chargées ou restaurées à tout moment. L'objectif est donc de déterminer l'identité de l'utilisateur qui a effectué la modification à un moment donné. L'objectif de tels systèmes est, d'une part, de coordonner l'accès commun aux fichiers par plusieurs utilisateurs et, d'autre part, de permettre à plusieurs branches de se développer (évoluer ou se séparer) simultanément.

Les systèmes de versionnement sont généralement utilisés dans le développement de logiciels et dans les systèmes de gestion de contenu dans les environnements professionnels. Les outils de gestion de version les plus connus sont Apache Subversion (SVN) et Git, qui peuvent être installés sur votre propre serveur ou loués auprès d'une société d'hébergement.

p. 3 Solution n°2

Il s'agit d'un outil open source basé sur Git qui permet de gérer les dépôts de code source et leurs versions, et propose toute une série de fonctionnalités autour d'un projet de développement.

GitLab est un outil qui peut être utilisé par une seule personne ou par une équipe, voire par une organisation ou une grande entreprise. C'est un outil complet, peut-être même complexe quand on est débutant.

Voici une liste des fonctionnalités proposée par GitLab :

- Un système de suivi de bug (ou des problèmes),
- Gestion des rôles des utilisateurs,
- Un tableau de bord interactif,
- Intégration et livraison continues,
- Hébergement de sites web statiques,
- Wikis.

p. 3 Solution n°3

Avant de commencer, il faut que Git soit déjà installé sur l'ordinateur en question. Afin de vérifier cela, il suffit de taper **« *$ git version* »** dans votre invite de commande. Cette commande permet de voir la version de Git installé.

Une fois cela fait, il faut créer un dépôt depuis de l'interface de GitLab :

- Cliquer sur « *Nouveau projet* ».
- Le nom du dépôt correspondant au nom du projet (le mieux c'est de donner un nom de projet en minuscules séparé par des tirets).
- Choisissez « *privé* », afin que le dépôt ne soit pas accessible que par les utilisateurs avec qui on décide de partager ce dépôt.
- Copier l'adresse HTTPS fournie dans la page du dépôt.

Ensuite, dans l'invite de commande, on télécharge le dépôt sur le disque dur avec la commande suivante :

$ git clone <url> (il faut remplacer <url> par l'adresse HTTPS de votre dépôt) puis entrez dans le dossier de ce dépôt : **$ cd nom_du_dépôt**.

Enfin, la dernière étape consiste à créer un commit. Pour cela, on commence par glisser tous les fichiers qu'on souhaite gérer dans le dossier. Ensuite, on fait un **$ git status.** Cette commande permet de vérifier que tous les fichiers ont bien été ajoutés au dossier, mais pas encore ajoutés dans le dépôt.

Ensuite, on fait un « ***$ git add.*** » pour ajouter tous ses fichiers dans l'espace de staging. Puis on exécute la commande **$ git commit -m** « ***ajout des fichiers*** » pour créer un commit à partir de l'espace de staging. Le texte fourni entre guillemets est libre. Il permet d'expliquer quelles modifications sont apportées au dépôt par ce commit.

Pour finir, on fait un **$ git push** pour uploader le commit sur le dépôt distant, hébergé sur le serveur GitLab.

p. 3 Solution n°4

Un audit ou test de pénétration (pentesting) consiste en des tests offensifs contre les mécanismes de défense existants dans l'environnement analysé. Il existe de nombreux cas où des organisations subissent des incidents qui auraient pu être évités si les mécanismes de protection avaient été renforcés à l'époque. Les incidents comprennent des événements tels que la fuite d'informations, l'accès non autorisé ou la perte de données, parmi beaucoup d'autres. L'analyse des mécanismes de protection doit être une tâche proactive permettant au pentesteur (personne qui réalise l'audit) de trouver leurs vulnérabilités et d'apporter une solution avant qu'un cybercriminel ne profite de cette faiblesse.

Ces processus permettent aux entreprises d'économiser l'argent et le temps nécessaires pour résoudre les problèmes futurs dus aux vulnérabilités des applications.

p. 3 Solution n°5

Les tests d'intrusion se divisent en trois grands groupes, en fonction de la méthode utilisée par le testeur et de son niveau de connaissances.

- Le Pentest BlackBox : l'auditeur fait une simulation de l'attaque en se plaçant dans la peau d'un vrai hacker. Cela implique que l'auditeur ne dispose que de peu ou pas d'informations sur la cible. Cette stratégie permet de définir de manière fiable les seuils de sécurité critiques d'une entreprise. Les pirates disposent généralement de peu de données sur le SI qu'ils cherchent à compromettre. Par conséquent, l'exploration du SI leur demande un certain temps, pendant lequel elles pourront réagir si l'entreprise cible a les moyens de le faire. Ils sont donc adaptés à la définition de scénarios dans lesquels une entité extérieure à l'entreprise tente de compromettre la sécurité. Cette approche ne permet pas à l'attaquant de se concentrer sur le contenu sensible du SI du client et ne convient pas à l'audit à court terme.
- Le Pentest WhiteBox : tests en boîte blanche : à la différence des tests en boîte noire, l'auditeur collabore étroitement avec le service informatique. Il a accès à toutes les informations relatives à la configuration du SI. Les tests WhiteBox five sont plus proches d'un audit informatique formel, mais ils donnent la possibilité d'approfondir la détection des vulnérabilités en accédant à toutes les couches du SI.
- Le Pentest Greybox : il s'agit d'une méthode intermédiaire utilisée pour bénéficier des avantages des boîtes noires et blanches. Dans ce cas, le pentester effectue des tests de pénétration à l'aide d'un nombre limité d'informations. Par exemple, il peut avoir rejoint l'entreprise en tant qu'employé d'un département sensible et disposer de son propre compte utilisateur. Au fur et à mesure qu'il progresse dans l'attaque, il obtient de nouvelles informations. Le test GreyBox est une stratégie optimale, car il permet de simuler plusieurs types d'attaques, y compris celles menées « *de l'intérieur* ». En fonction des droits qui lui sont attribués, le pentester peut élaborer des scénarios d'attaque provenant de membres ou d'anciens employés de l'entreprise, ou même de prestataires de services externes.

p. 3 Solution n°6

Afin de réaliser un Pentest, il faut suivre les étapes suivantes.

- La collecte d'informations : elle implique la collecte de données et d'informations sur la cible. Les sources d'information peuvent varier en fonction de la nature du test de pénétration. Ce peuvent être des sources externes ouvertes à tous les utilisateurs, comme les moteurs de recherche, ou des informations fournies par l'entreprise elle-même.
- Faire un inventaire sur l'ensemble des actifs : cette phase permet de se concentrer sur les éléments considérés comme vitaux et délicats.
- Recherche de vulnérabilités : elle permet d'analyser les vulnérabilités des applications, des sites web et des systèmes à partir des données recueillies.
- Exploitation : la phase de développement représente l'application du travail effectué précédemment. Les pentester essayeront de s'introduire par chacun des défauts mis en évidence afin d'établir un contrôle sur le système d'information du client.
- Élévation de privilèges : elle permet au testeur d'acquérir une plus grande autorité afin de prendre en charge le travail des administrateurs système et d'effectuer davantage de tâches.
- Maintien d'accès : le défi consiste maintenant à obtenir et à maintenir ces nouveaux privilèges. À ce stade, le pentester doit agir avec une extrême prudence, effaçant ses traces pour ne pas éveiller les soupçons de l'équipe technique.
- Propagation : il s'agit notamment d'étendre les dommages d'un poste de travail à d'autres machines et serveurs du parc informatique.
- CleanUp : à la fin de l'audit, les traces laissées par les auditeurs sur le SI sont nettoyées. L'objectif est de rendre le SI dans son état d'origine.
- Le reporting : l'étape la plus importante des tests de pénétration. Le testeur doit fournir à son client un rapport d'audit écrit expliquant sa méthodologie et identifiant les éventuelles failles et vulnérabilités du système d'information. Il doit inclure :
    - Une synthèse technique
    - Les préconisations de remédiations
    - Une synthèse managériale
    - Le plan d'action