M2-TIW4 sécurité des systèmes d'informations
Contrôle continu final – durée 1h30

Master Technologie de l'Information, promotion 2017 – 2018

Mercredi 14 mars 2018

Aucun document autorisé. Le barème est indicatif. La concision, la précision et la clarté des réponses font partie intégrante de l'évaluation. Ne pas rendre le sujet avec la copie.

Exercice 1 : Pot-pourri (/5)

Pour chaque question, donnez simplement le réponses ordonnées sur votre copie, e.g., « C, A, A, B » pour la première ou « vrai, vrai, vrai, faux » pour les autres.

1. On donne trois catégories de la classification SANS : (A) *Insecure Interaction Between Components*, (B) *Risky Resource Management* et (C) *Porous Defenses*. Pour chacune des erreurs suivantes, indiquer la catégorie à laquelle elle se rattache. (/1)
   - ☐ *Use of a One-Way Hash without a Salt* ;
   - ☐ *Incorrect Calculation of Buffer Size* ;
   - ☐ *URL Redirection to Untrusted Site* ;
   - ☐ *Use of a Broken or Risky Cryptographic Algorithm* ;
2. Donner la ou les réponses correctes : le contrôle d'accès... (/1)
   - ☐ Discrétionnaire ne suppose pas d'autorité qui régit les droits de tous ;
   - ☐ Mandataire ne suppose pas d'autorité qui régit les droits de tous ;
   - ☐ Discrétionnaire est traditionnellement utilisé dans les systèmes de fichiers Linux ;
   - ☐ Mandataire est traditionnellement utilisé dans les systèmes de fichiers Linux.
3. Avec la hiérarchie *Très Secret Défense* (TS), *Secret Défense* (S), *Confidentiel Défense* (C) et *Non-classifié* (U), on suppose que l'on a deux utilisateurs u_0 et u_1 accrédités respectivement C et S ainsi que deux fichiers f_0 et f_1 auxquels sont associés les niveaux respectifs TS et C. Donner la ou les réponses correctes : (/1)
   - ☐ u_0 peut lire et écrire dans f_1 ;
   - ☐ u_1 peut écrire dans f_0 ;
   - ☐ u_1 peut écrire dans f_1 ;
   - ☐ si on accrédite u_1 à C, les droits restent les mêmes.
4. Commenter la citation suivante de Kevin Mitnick du point de vue de la sécurité « *Don't rely on network safeguards and firewalls to protect your information. Look to your most vulnerable spot. You'll usually find that vulnerability lies in your people.* » (/2)

Exercice 2 : Protocole de Woo-Lam (/5)

Le protocole de Woo-Lam est un protocole où A contacte B en s'appuyant sur un serveur S qui partage un secret avec chacun des participants. Le protocole est le suivant :

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_B$
3. $A \rightarrow B : \langle N_B \rangle_{K_{AS}}$
4. $B \rightarrow S : \langle A, \langle N_B \rangle_{K_{AS}} \rangle_{K_{BS}}$
5. $S \rightarrow B : \langle A, N_B \rangle_{K_{BS}}$

1. Expliquer quels sont les éléments qui apparaissent dans le contenu des messages de ce protocole : A, B, N_B, K_{AS} et K_{BS} (/2)
2. De quel type de protocole s'agit-il, à quoi sert-il ? (/1)
3. On considère l'échange ci-après, où un tiers malin M se fait passer pour A (notation $M(A)$) puis pour S (notation $M(S)$) auprès de B. Quel est le problème soulevé [1] ? (/2)
   1. $M(A) \rightarrow B : A$
   2. $B \rightarrow M(A) : N_B$
   3. $M(A) \rightarrow B : N_B$
   4. $B \rightarrow M(S) : \langle A, N_B \rangle_{K_{BS}}$
   5. $M(S) \rightarrow B : \langle A, N_B \rangle_{K_{BS}}$

Exercice 3 : Gestion des mots de passes `PhP/Mysql` (/8)

On lit dans le sujet de TP d'un camarade l'énoncé suivant :

> *En regardant dans la base de données, on s'aperçoit que les utilisateurs sont enregistrés avec leurs mots de passe en clair ce qui constitue une faille de sécurité majeure. Modifier la récupération du mot de passe dans le fichier `inscription.php` pour que le mot de passe soit crypté (hachage cryptographique MD5) dans la base de données.*

On y lit également l'exemple suivant d'utilisation de `mysqli` :

```
<?php
$salaire_max = /* valeur obtenue depusi une saisie utilisateur */ ;
$requete = "SELECT nom,salaire FROM employe WHERE salaire <= $salaire_max";
$resultat = mysqli_query($connexion, $requete);
...
?>
```

1. Quelle est la *faille de sécurité majeure* dont il est question dans l'énoncé ? Illustrez avec des risques. (/2)
2. Expliquez quel est le problème posé par la méthode suggérée de gestion des mots de passes par *hash* MD5. (/2)
3. Reformulez la consigne avec une meilleure pratique en matière de sécurité. (/2)
4. Quant au code PhP d'exemple d'utilisation de `mysqli`, que proposeriez vous, dans une perspective d'amélioration de la sécurité ? Justifiez. (/2)

1. On supposera que B ne peut pas faire la différence entre un message chiffré et un message aléatoire

Exercice 4 : Problème de modélisation et de migration des droits d'accès (/22)

On s'intéresse à la conception d'un système de contrôle d'accès pour une application de gestion documentaire. Un extrait du schéma de la base de données métier est donné ci-dessous. Les clefs primaires sont soulignées et les clefs étrangères précédées du symbole #. Un utilisateur qui est membre d'une équipe a comme rôle soit « responsable », soit « membre », soit « invité ».

Utilisateur(idUsr, nom, prenom)
Equipe(idEqp, nom)
Document(idDoc, #idCreateur, url, creationDate)
Membre(#idEqp, #idUsr, role)
Associe(#idEqp, #idDoc)

Une requête de création d'un document est exprimée par un utilisateur qui fournit une liste des équipes auxquelles il souhaite associer le document. Lors de l'ajout dans la table Document les attributs id, url sont calculés automatiquement, on suppose pour cela qu'on dispose d'une fonction « *NouveauDoc*() » qui renvoie la paire de valeurs $\langle \text{id}, \text{url} \rangle$. Une requête d'accès à un document est exprimée par un utilisateur qui indique à quel document il souhaite accéder en précisant l'action à effectuer, les actions étant $\{\text{read}, \text{write}, \text{delete}\}$.

La politique de contrôle à prendre en compte dans l'exercice est définie en langue naturelle ci-après. Dans un premier temps, on va s'intéresser à modéliser cette politique sur la base de données métier en implémentant un moniteur de contrôle d'accès.

> *Un utilisateur peut effectuer toutes les opérations sur les documents dont il est le créateur. Un membre d'une équipe peut accéder en lecture à tous les documents associés à son équipe, quel que soit son rôle. Le responsable d'une équipe peut écrire dans tous les documents associés à son équipe. Lors de la création d'un document, son créateur ne peut associer le nouveau document qu'aux équipes dont il est « responsable » ou « membre ».*

Dans un second temps, on va supposer qu'il existe un produit logiciel de contrôle d'accès de type RBAC et on va traduire la politique dans ce système. Pour rappel, le schéma typique de la base RBAC est donnée ci-dessous, les utilisateurs étant les mêmes que ceux de la base métier. Notons qu'on ne considère ni session, ni hiérarchie de rôles ni contraintes d'exclusion entre rôles.

Role(idRole, ...)
Permission(#idDoc, action)
URA(#idRole, #idUsr)
PRA(#idRole, #idDoc, action)

1. Un utilisateur peut-il disposer de plusieurs rôles dans une équipe ? Un utilisateur peut-il être responsables de plusieurs équipes ? Un document est-il toujours associé à au moins une équipe ? Justifiez. (/2)
2. Formellement, une création de document est une commande « *CreerDoc*(*idUsr*, [*listeEquipes*]) » qui va ajouter le document dans la base métier et renvoyer l'identifiant du document créé. Définir cette commande[2]. (/3)
3. Formellement, une demande d'accès à un document est une commande « *Acces*(*idUsr*, *idDoc*, *action*) » où action $\in \{\text{read}, \text{write}, \text{delete}\}$ qui va renvoyer un booléen selon que l'accès soit autorisé ou non. Définir cette commande. (/3)
4. Si un utilisateur ne précise aucune équipe lors de la création d'un document, quels sont les utilisateurs qui disposent de droits sur ce document ? Justifiez votre réponse sur vos définitions précédentes et proposez un test unitaire correspondant. (/2)
5. Formellement, une vérification d'accès RBAC est une commande « *RBAC*(*idUsr*, *idDoc*, *action*) » qui renvoie un booléen selon que l'accès soit autorisé ou non. Définir cette commande. (/2)
6. On s'intéresse enfin à la modélisation des droits dans le modèle RBAC. Pour cela, vous devez identifiez la liste des rôles et les affectations de rôles aux utilisateurs (relation URA) et de permissions aux rôles (relation PRA). Précisez la relation Role(<u>idRole</u>, ...) en définissant des attributs et clefs supplémentaires utiles et définissez une commande « *MigrateRoles*() » qui lit la base de données métier et peuple la table Role. (/2)
7. Définissez une commande « *MigrateURA*() » qui lit la base de données métier et peuple la table URA. (/3)
8. Définissez une commande « *MigratePRA*() » qui lit la base de données métier et peuple la table PRA. (/3)
9. La migration est dite *correcte* si une commande « *Acces*(*idUsr*, *idDoc*, *action*) » est autorisée *si et seulement si* « *RBAC*(*idUsr*, *idDoc*, *action*) » est autorisée. Justifiez qu'une migration est dite correcte si aucun droit n'est perdu et aucun droit n'est créé. Donnez des éléments justifiant que la migration que vous avez proposée est bien correcte. (/2+)

2. Dans toutes les questions, utilisez du pseudo-code le plus clair et lisible possible. Pour les parties qui font accès à la base de données, utilisez de préférence une syntaxe à la SQL, e.g., *let es* = `SELECT idEqp FROM Equipe;` ou une notation ensembliste, e.g., *let es* = $\{idEqp \mid \exists n.\text{Equipe}(\text{idEqp}, \text{n})\}$ ou encore *for* $\langle e, n\rangle \in$ Equipe pour une boucle.