CORRIGÉ NON OFFICIEL

BTS Services Informatiques aux Organisations

E6 - Cybersécurité des services informatiques (spécifique à l'option)

Session 2026

Proposition de corrigé rédigée par FormaV pour s'entraîner. Ce n'est pas le corrigé officiel de l'examen et elle ne remplace pas les éléments de correction officiels.

Ce document a été mis en ligne par l'organisme FormaV®.

Toute reproduction, représentation ou diffusion, même partielle, sans autorisation préalable, est strictement interdite.

Pour consulter tous nos sujets d'annales et corrigés du BTS SIO, veuillez visiter :
coursbtssio.fr/annales-sujets-et-corriges

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS – Option : Solutions logicielles et applications métiers

SESSION 2026

U7 – CYBERSÉCURITÉ DES SERVICES INFORMATIQUES

PROPOSITION DE CORRIGÉ

Proposition de corrigé établie par FormaV à partir du sujet, non officielle. Elle ne préjuge pas des consignes nationales de correction données aux jurys.

DOSSIER A – ANALYSE DES RISQUES LIÉS À L'APPLICATION GESSYCA — 33 points

Question A1.1

Justifier l'évaluation du récit utilisateur 28 pour chacun des critères de sécurité.

Le récit 28 concerne un gestionnaire chargé de la comptabilité qui télécharge les données des ventes du mois en cours, contenant les informations client.

Disponibilité (*, modéré) : le téléchargement est une opération ponctuelle. Son indisponibilité temporaire ne bloque ni la prise de commande ni la confection des produits, d'où un niveau modéré.

Intégrité (, important)** : les données de ventes servent à la comptabilité. Une altération des montants, des produits ou des clients fausserait les comptes de l'association et les décisions qui en découlent.

Confidentialité (, important)** : le fichier contient les informations des clients, qui sont des données à caractère personnel. Leur divulgation constituerait une violation de données personnelles au sens du RGPD et exposerait les clients.

Preuve (-, sans objet) : il s'agit d'une extraction de consultation, sans effet sur l'engagement des commandes ou sur les paiements. Ce récit ne suppose pas de démontrer juridiquement une action ou un engagement.

Question A1.2

Justifier, pour chaque photo de cette commande, de l'utilité ou non d'un tel document, selon le RGPD.

Une photographie permettant d'identifier une personne est une donnée à caractère personnel (article 4 du RGPD). Son utilisation pour fabriquer un objet personnalisé constitue un traitement qui doit reposer sur une base légale. Ici, la base légale retenue est le consentement de la personne concernée (article 6.1.a du RGPD). Ce consentement doit être libre, éclairé, spécifique, univoque et retirable (article 7 du RGPD).

Photo des enfants (tasses) : les enfants sont identifiables et, étant mineurs, appellent une attention particulière. Le document de consentement est utile. Il doit être signé par les titulaires de l'autorité parentale, la cliente pouvant consentir pour ses enfants uniquement si elle en a la qualité. Le document permet de prouver ce consentement et de respecter le principe de responsabilité (accountability).

Photo du paysage (tapis de souris) : aucune personne identifiable n'apparaît sur la photo. Il ne s'agit donc pas d'une donnée à caractère personnel, le RGPD ne s'applique pas et le document de consentement est inutile au regard du RGPD. Si une personne identifiable figurait sur ce paysage, la photo relèverait de la même exigence de consentement que la première.

Hors RGPD, le droit à l'image (article 9 du Code civil) protège la personne contre la reproduction de son image sans son accord, ce qui renforce l'utilité du document pour la photo des enfants.

Question A.1.3

Indiquer le traitement à réaliser sur les images à caractère personnel une fois la commande livrée.

Une fois la commande livrée, les images à caractère personnel doivent être **supprimées** (effacement) de l'application et de l'espace de stockage, y compris des copies de sauvegarde selon la politique de l'association. Ce traitement découle du principe de limitation de la conservation (article 5.1.e du RGPD) : les données ne sont conservées que le temps nécessaire à la finalité, ici la fabrication de la commande. La conservation n'est admise que si le client y a consenti pour une nouvelle commande ou si une obligation légale l'impose.

Question A2.1

a) Expliquer pourquoi le système actuel des messages d'erreur pose problème. b) Proposer une autre solution plus sécurisée. c) Donner deux exemples de types d'attaque pouvant profiter du fait que les tentatives de connexion ne sont pas limitées.

a) Les deux messages distincts, « identifiant non reconnu » et « mot de passe incorrect », permettent à un attaquant de savoir si un identifiant existe dans l'application. Il s'agit d'une énumération des comptes : l'attaquant peut ensuite concentrer ses tentatives sur les comptes valides.

b) Il faut afficher un message unique et générique, identique quel que soit le cas, par exemple « Identifiant ou mot de passe incorrect ». Le temps de réponse doit également être uniforme. Cette solution peut être complétée par la limitation des tentatives et par une authentification multifacteur.

c) Deux types d'attaque peuvent profiter de l'absence de limitation des tentatives : l'attaque par force brute (essais systématiques de mots de passe) et le bourrage d'identifiants (credential stuffing), qui consiste à rejouer des couples identifiant-mot de passe volés sur d'autres services. La pulvérisation de mots de passe (password spraying) est également acceptée.

Question A3.1

Décrire les contraintes imposées aux utilisateurs de l'application en matière de gestion des mots de passe.

Les contraintes imposées aux utilisateurs sont les suivantes :

- le mot de passe comporte au moins 8 caractères (fonction verifierStructMotDePasse()) ;
- il contient au moins un chiffre, une minuscule, une majuscule et un caractère spécial (fonction verifierStructMotDePasse()) ;
- pour un utilisateur ayant le rôle administrateur, le mot de passe doit être modifié tous les trois mois : la fonction verifierDateMotDePasse() vérifie, lors de la connexion, que la date de la dernière modification n'est pas antérieure à trois mois ;
- l'utilisateur ne doit pas réutiliser l'un de ses trois derniers mots de passe : le déclencheur before_update_utilisateur interdit la mise à jour si le nouveau mot de passe correspond à l'une des trois dernières valeurs enregistrées dans Histo_Mdp.

Remarque : la comparaison des empreintes dans le déclencheur ne détecte pas réellement la réutilisation, car password_hash() produit une empreinte différente pour un même mot de passe grâce au sel aléatoire. Le contrôle devrait vérifier le mot de passe avec password_verify() sur chaque ancienne empreinte.

Question A3.2

a) Décrire la façon dont le salage a été implémenté dans le code. b) Définir la technique du salage. c) Préciser le type d'attaque auquel le salage permet, entre autres, de faire face.

a) Dans la fonction inscrireUtilisateur(), le mot de passe est haché par l'appel *password_hash($password, PASSWORD_DEFAULT)*. Le développeur n'écrit aucun sel explicitement : la fonction génère automatiquement un sel aléatoire et sécurisé, l'intègre dans l'empreinte produite, et l'empreinte complète est stockée dans la colonne mdp de la table Utilisateur. Le choix de PASSWORD_DEFAULT permet de passer automatiquement à un algorithme plus fort lors des mises à jour de PHP.

b) Le salage consiste à ajouter au mot de passe une valeur aléatoire, appelée sel (salt), avant de le hacher. Le sel est généralement unique pour chaque utilisateur. Deux utilisateurs ayant le même mot de passe obtiennent alors deux empreintes différentes, et les empreintes précalculées deviennent inutilisables.

c) Le salage permet notamment de faire face aux attaques par tables arc-en-ciel (rainbow tables), qui reposent sur des empreintes précalculées de mots de passe courants.

DOSSIER B – SÉCURISATION DES ACCÈS À L'APPLICATION GESSYCA 33 points

Question B1.1

Repérer dans les journaux d'accès du serveur de base de données de l'application GESSYCA les éléments qui pourraient laisser penser que l'association a pu faire l'objet d'une cyber-attaque.

Le journal /var/log/mysql/mysql.log montre les éléments suivants, pour la connexion 294 du 11 janvier 2026 à 23 h 37 min 45 s (date au format AAMMJJ) :

- la connexion utilise le compte *root* (compte aux privilèges complets, utilisé par l'application d'après f_bdd.php) ;
- la connexion a lieu en dehors des heures de travail, à un horaire où aucun membre de l'association n'est censé intervenir ;
- une requête UPDATE modifie l'état de toutes les commandes « En cours » en « Clôturée » ;
- cette mise à jour fait passer directement les commandes de « En cours » à « Clôturée », en sautant les états « Validée », « En Production », « Terminée » et « Réglée », ce qui viole le processus défini dans le document 2 ;
- une requête SELECT repère d'abord toutes les commandes « En cours », puis une requête UPDATE les modifie en masse dans la même session, juste avant la déconnexion (Quit).

Question B1.2

Détailler les impacts de cet acte malveillant pour l'association LES OLINS.

Les commandes « En cours » ont été clôturées sans avoir été validées, confectionnées, livrées ni réglées. Cet acte entraîne les impacts suivants :

- des commandes ne seront pas produites, ce qui retarde ou empêche la livraison aux clients et nuit à la disponibilité du service ;
- les états des commandes ne correspondent plus à la réalité : l'intégrité des données est atteinte et l'association ne peut plus suivre son activité ;
- la facturation et l'encaissement sont perturbés, avec un risque de perte de chiffre d'affaires ;

- la confidentialité de la base de données est menacée, puisque l'accès s'est fait avec un compte administrateur ;
- l'association doit mobiliser du temps pour restaurer les données, identifier les commandes concernées et informer les clients, ce qui engendre des coûts et peut entamer la confiance des clients et des partenaires.

Question B1.3

Citer l'une des infractions qui peut être retenue contre la personne qui a modifié le statut des commandes de façon malveillante.

Il peut être retenu la modification frauduleuse de données dans un système de traitement automatisé de données, prévue à l'article 323-3 du Code pénal. Sont également envisageables l'accès frauduleux à un système de traitement automatisé de données (article 323-1) et l'entrave à son fonctionnement (article 323-2).

Question B1.4

Citer le document qui précise les risques encourus par le membre qui a perdu son carnet.

Le document visé est la charte informatique (charte d'utilisation des moyens informatiques), qui fixe les règles de bon usage des identifiants et les sanctions encourues en cas de manquement, notamment la divulgation ou la mauvaise conservation des identifiants de connexion.

Question B2.1

Justifier, en matière de preuve, le choix de désactiver les utilisateurs plutôt que de les effacer.

Désactiver un compte plutôt que de l'effacer conserve la ligne de la table Utilisateur. Les commandes, l'historique des mots de passe (Histo_Mdp) et les traces sont liés à cet utilisateur par des clés étrangères (idUtilisateur). Il reste donc possible de savoir quel membre a saisi ou modifié une commande, à quelle date, ce qui assure la traçabilité et la preuve des actions passées. L'effacement aurait supprimé ces liens, ou aurait imposé de les modifier, et aurait fait disparaître des éléments de preuve.

Question B2.2

Identifier les trois éléments de sécurité mis en place dans le traitement de la désactivation d'un membre.

Les trois éléments de sécurité sont :

1. le jeton CSRF : le formulaire contient un jeton généré par csrf_token.php, et validerCsrfToken() le compare au jeton de la session avec hash_equals(), ce qui empêche la soumission d'une requête forgée par un site tiers ;
2. la validation de la date : la fonction validerDate() vérifie que la date est valide et respecte exactement le format Y-m-d ;
3. la requête préparée : la mise à jour est paramétrée avec PDO (prepare, bindValue), ce qui protège contre l'injection SQL.

Sont également acceptés : le contrôle des champs obligatoires (empty), le filtrage des saisies par trim() et htmlspecialchars().

Question B2.3

Compléter le code de la fonction desactiverUtilisateurs() pour désactiver les utilisateurs appropriés.

Les utilisateurs appropriés sont ceux dont la date de désactivation est égale ou antérieure à la date actuelle et qui sont encore actifs. Le code complété est le suivant :

```
$dateActuelle = date('Y-m-d');
$resultat = false;
try {
    $cnx = connexionPDO();
    $sql = "UPDATE Utilisateur SET estActif = FALSE
        WHERE dateDesactivation <= :dateActuelle AND estActif = TRUE";
    $req = $cnx->prepare($sql);
    $req->bindValue(':dateActuelle', $dateActuelle, PDO::PARAM_STR);
    $resultat = $req->execute();
} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage();
    die();
}
return $resultat;
```

Une valeur NULL dans dateDesactivation ne satisfait pas la condition, donc les comptes sans date de désactivation ne sont pas touchés.

Question B2.4

Identifier la faiblesse principale du script f_bdd.php.

La faiblesse principale est que la connexion à la base de données est établie avec le compte *root*, qui dispose de tous les privilèges sur le serveur. Ce choix ne respecte pas le principe du moindre privilège : en cas de compromission de l'application, l'attaquant obtient des droits complets sur le serveur de base de données.

Question B2.5

Proposer une alternative plus sécurisée.

Les identifiants doivent être sortis du code source et stockés hors de l'arborescence web, dans un fichier de configuration à droits restreints (lisible uniquement par le serveur web) ou dans des variables d'environnement, voire dans un gestionnaire de secrets. Le compte utilisé par l'application doit être un compte dédié, disposant uniquement des droits nécessaires (SELECT, INSERT et UPDATE sur les seules tables de DB_GESSYCA, sans droits d'administration du serveur), et le mot de passe du compte root, exposé dans le code, doit être immédiatement changé.

Question B2.6

Proposer une requête qui permette de s'assurer du bon fonctionnement du traitement.

La requête recherche les comptes dont la date de désactivation est atteinte mais qui sont restés actifs. Si le traitement a bien fonctionné après la panne, elle ne doit renvoyer aucune ligne :

```
SELECT id, login, dateDesactivation, estActif
FROM Utilisateur
WHERE dateDesactivation <= CURDATE() AND estActif = TRUE;
```

DOSSIER C – SÉCURISATION DE L'APPLICATION DE GESTION DES COMMANDES — 14 points

Question C1.1

Indiquer quel critère de la sécurité de l'information la base de données ne respecte pas concernant l'état de la commande.

La base de données ne respecte pas le critère de **preuve** (traçabilité) : seul l'état courant de la commande est conservé (champ idEtat de Commande). Il est donc impossible de savoir qui a fait passer une commande d'un état à un autre, ni à quelle date. Les changements d'état ne sont ni tracés ni imputables à un utilisateur, ce qui empêche toute justification ou contestation ultérieure.

Question C1.2

Proposer les modifications à réaliser pour répondre à ces exigences. Seuls les éléments du schéma existant qui sont concernés par l'évolution seront repris dans le formalisme de votre choix.

Il faut conserver l'état courant dans la table Commande (champ idEtat) et enregistrer chaque changement d'état dans une nouvelle table d'historique, qui associe la commande, l'état, l'utilisateur responsable et la date du changement, avec un commentaire facultatif.

Schéma relationnel modifié :

Historique_Etat (idHisto, dateChangement, commentaire, idCommande, idEtat, idUtilisateur)
Clé primaire : idHisto
Clés étrangères : idCommande en référence à id de Commande ; idEtat en référence à id de Etat ; idUtilisateur en référence à id de Utilisateur

Les attributs sont : dateChangement de type DATETIME, non nul ; commentaire de type TEXTE, facultatif.

Le diagramme de classes doit intégrer une classe HISTO_ETAT (dateChangement, commentaire) reliée à COMMANDE (1 à plusieurs, sens 1 vers n), à ETAT et à UTILISATEUR (chacune de type 1 à plusieurs).

Pour les commandes passées, il est possible d'insérer une ligne initiale pour chaque commande : la date de la commande (dateCom) et l'utilisateur ayant créé la commande (idUtilisateur) servent de point de départ. Les dates des changements antérieurs n'étant pas enregistrées, elles ne peuvent pas être reconstituées.

Afin que chaque modification soit tracée, l'application insère une ligne dans Historique_Etat (idCommande, idEtat, idUtilisateur de la session, date du jour, commentaire saisi) dans la même transaction que la mise à jour de Commande.idEtat ; à défaut, un déclencheur AFTER UPDATE peut utiliser une variable de session SQL (@idUtilisateur) positionnée par l'application.

Question C2.1

a) Donner la cause probable de l'erreur obtenue. b) Préciser le risque de sécurité OWASP auquel ce mécanisme est censé répondre.

a) Le test (document C3), une requête GET sur https://api.insee.fr/api-sirene/3.11/siren/830609764, renvoie le code 401-Unauthorized ({"message": "Unauthorized"}). D'après le document C1, ce code signifie que la clé d'API est manquante ou invalide : la requête GET a été envoyée sans transmettre dans l'en-tête (Header) la clé (API KEY) obtenue après la souscription de l'application à Sirene open data (document C2), ou avec une clé erronée ou expirée.

b) Le mécanisme de clé d'API sert à authentifier l'application appelante. Il répond au risque OWASP API02 : Authentification rompue.

Question C2.2

Nommer le type d'attaque qui provoque cette erreur.

L'erreur 429-Too Many Requests signale que le quota d'interrogations est dépassé. Elle est provoquée par une attaque par déni de service (DoS), réalisée par l'envoi massif de requêtes vers l'API afin de la saturer.

Sont également acceptés : déni de service distribué (DDoS) si les requêtes proviennent de plusieurs sources.

TABLEAU RÉCAPITULATIF DU BARÈME

Partie	Points
Dossier A – Analyse des risques liés à l'application GESSYCA	33
Dossier B – Sécurisation des accès à l'application GESSYCA	33
Dossier C – Sécurisation de l'application de gestion des commandes	14
Total	**80**

Continue à t'entraîner

Retrouve tous les sujets et corrigés du BTS SIO, classés par épreuve et par année :
coursbtssio.fr/annales-sujets-et-corriges

Pour réviser tout le programme avant l'examen, les Fiches de Révision du BTS SIO reprennent l'essentiel de chaque épreuve :
coursbtssio.fr/fiches-de-revision

Copyright © 2027 FormaV®. Tous droits réservés.

FormaV® est une marque déposée auprès de l'INPI sous le numéro 5227897.

Ce document a été élaboré par FormaV® avec le plus grand soin afin d'accompagner chaque apprenant vers la réussite de ses examens. Son contenu (textes, graphiques, méthodologies, tableaux, exercices, concepts, mises en forme) constitue une œuvre protégée par le droit d'auteur.

Toute copie, partage, reproduction, diffusion ou mise à disposition, même partielle, gratuite ou payante, est strictement interdite sans accord préalable et écrit de FormaV®, conformément aux articles L.111-1 et suivants du Code de la propriété intellectuelle. Dans une logique anti-plagiat, FormaV® se réserve le droit de vérifier toute utilisation illicite, y compris sur les plateformes en ligne ou sites tiers.

En utilisant ce document, vous vous engagez à respecter ces règles et à préserver l'intégrité du travail fourni. La consultation de ce document est strictement personnelle.

Merci de respecter le travail accompli afin de permettre la création continue de ressources pédagogiques fiables et accessibles.