Stratégie de sécurisation de la plateforme en ligne pire2pire.com

1 - INTRO

In today's increasingly interconnected world, securing information systems and data is a critical priority for all organizations. In this regard, it's crucial to determine the targets, assess the level of protection they require and understand the repercussions in case of a security breach.

The repercussions of security flaws can be particularly frustrating for those involved including:

- The user: Potential target of identity or information theft which can result in a significant loss of trust in the application.
- The application owner: A security breach could cause service interruption and compliance violation leading to legal penalties and financial loss.
- The application developer: Should there be flaws in security, it could impact the organization's reputation, result in potential lawsuits, and lead to financial losses associated with remediation effort.

This security strategy seeks to establish a comprehensive framework outlining the necessary steps to implement robust protections against the risks we encounter. In this document, we will address each layer involved, identify potential risks, and outline appropriate countermeasures.

The security strategy encompasses several critical layers, such as:

- Transversal security rules and good practices
- Frontend vulnerabilities
- Api vulnerabilities
- Database injections and malware threats

This strategy was designed with a keen focus on optimizing the balance between ergonomics and safety

☺ 👍 ☹

The rigorous implementation of this strategy will not only strengthen the trust of our stakeholders but also protect all our business processes from potential compromise.

2 - RÈGLES DE SÉCURITÉ TRANSVERSALES

La présente section a pour but de présenter le socle de règles et de bonnes pratiques qui serviront à construire les fondations d'une application sécurisée. Seront présentées dans cette partie les dispositions utiles à toutes les couches de l'application et les éléments de sécurité situés à mi-chemin entre deux couches. Seront détaillés également les concepts plus généraux d'hygiène de sécurité qui ne s'apparente pas spécialement à une couche ou une menace particulière.

2.1 - Défense en profondeur

La défense en profondeur repose sur l'implémentation de multiples mesures de protection indépendantes pour chaque menace identifiée. Ce principe s'applique plus efficacement lorsque le système est segmenté en unités distinctes, avec des interactions bien définies et des mécanismes de sécurité propres à chacune. Une approche incorrecte consisterait à concentrer toutes les protections au niveau de l'entrée du système, laissant ainsi les composants internes vulnérables en cas de faille en amont. Chaque composant logiciel de l'application, ainsi que l'infrastructure d'hébergement, doit idéalement contribuer à la sécurité globale du système.

2.2 Moindre privilège

Ce principe consiste à accorder aux composants et utilisateurs du système uniquement les permissions nécessaires à leur fonctionnement, afin de réduire les risques de vol, d'altération ou de destruction de données en cas de compromission de certains éléments.

2.2.a RBAC(Role Based Access Control)

Le contrôle d'accès basé sur les rôles (RBAC) attribue des permissions aux utilisateurs selon leur rôle au sein de l'organisation. Chaque rôle regroupe des droits spécifiques, limitant ainsi l'accès aux données sensibles uniquement à ceux qui en ont besoin pour leur fonction. Cette approche réduit les risques de sécurité et facilite la gestion des accès, renforçant ainsi la protection des données. Il est à noter que la gestion des autorisations des groupes authentifiés doit être complétée d’une politique de gestion des accès pour les utilisateurs non-authentifiés.

2.3 Réduction de la surface d'attaque

La réduction de la surface d’attaque consiste à limiter l'exposition des services, des points d'entrée et des composants logiciels non essentiels pour diminuer les risques d'exploitation des vulnérabilités. Cela implique de désactiver les services inutiles, de restreindre les accès aux interfaces, et d'éliminer les dépendances superflues dans le développement logiciel. En parallèle, il est crucial de segmenter le réseau, d'utiliser des mécanismes de validation et de journaliser les accès pour détecter toute activité suspecte. En intégrant ces pratiques, on renforce la sécurité globale du système.

2.4 Sécurité des échanges de données

La sécurité des échanges de données est cruciale pour garantir que les informations soumises ou reçues ne soient ni interceptées ni modifiées. Ce principe est particulièrement important pour les données personnelles, conformément à la loi informatique et libertés et au RGPD. L’objectif est d’assurer que les données proviennent bien du site consulté et non d’une source frauduleuse. Pour cela, l'utilisation du protocole HTTPS est essentielle, car il chiffre les communications et protège ainsi la confidentialité, l'intégrité et l'authenticité des données échangées sur le Web.

2.4.a Utilisation de TLS

La mise en place de TLS (Sécurité de la Couche de Transport) permet d'accroître le niveau de sécurité et de confidentialité lors des échanges client-serveur en satisfaisant trois principes clés :

- S'assurer de l'authenticité du serveur contacté (via émission d'un certificat d'authentification asymétrique)
- S'assurer de la confidentialité, en chiffrant les données qui transitent
- S'assurer de l'intégrité des données (via hashage)

2.4.b Utilisation de HSTS

TLS sera utilisé conjointement avec HSTS (HTTPS Strict Transport Security) afin de garantir que absolument tous les échanges se font par HTTPS. Cette restriction a pour but d'empêcher un individu malveillant d'intercepter la communication et la rediriger vers une version non sécurisée du site (HTTP), cette attaque est connue sous le nom de Man-In-The-Middle.

2.4.c Utilisation de CT

Mise en place de Certificate Transparency afin de garantir un environnement sain en s'assurant que lors des échanges client-serveur le certificat utilisé est bien référencé dans les journaux tenus à jour par les autorités de certification.

2.5 Conformité du contenu présenté

L'objectif de la conformité du contenu d'un site ou d'une application web est de garantir que le navigateur affiche correctement l'application selon l'intention du développeur, en empêchant toute altération malveillante ou imprévue durant l'échange des données.

2.6 Audit

Au-delà de toutes les mesures prises en termes de sécurité, il est à noter qu'elles sont à accompagner de mesures intra ou post conception telles que :

- Les outils de détection de dépendances vulnérables
- Les outils de monitoring
- Les tests de pénétration

Rentre également dans cette catégorie la mise en place de journaux d'événements permettant de tracer les actions effectuées et les dysfonctionnements du système.

2.7 Politique de mots de passe

Il convient d'observer quelques règles spécifiques afin de garantir une sécurité optimale des mots de passe des utilisateurs. Dans cette optique :

- Une vérification devra être effectuée avant de valider un nouveau mot de passe pour attester de :
  - L'utilisation correcte de caractères minuscules, de majuscule, de chiffre et de caractère spécial.
  - La longueur minimale est respectée en adéquation avec le type de compte.
  - Une limite devra de longueur de mot de passe devra néanmoins être appliquée au vu de la nature d'application web du système dans le but de se prémunir d'éventuelles attaques par déni de service.
  - L'absence de données personnelles telles que le nom, l'âge ou la date de naissance.
- Les mots de passe ne devront pas être stockés en clair dans la base de données, on leur préférera une utilisation de la méthode hashage – salage.
- La durée des sessions authentifiées sera limitée dans le temps, pour certaines actions (comme la suppression d'un cours), une reconnexion sera exigée.

- Le nombre de tentatives de connexions devront être limitées par un captcha au bout de quelques tentatives puis verrouillées dans le temps afin de se prémunir des tentatives d’assaut par force brute
- La demande de réinitialisation de mot de passe devra s’effectuer via un lien à durée de vie limitée envoyé sur le mail associé au compte.
- Lors de la création de mot de passe, ajouter un texte de sensibilisation aux bonnes pratiques de création d’un mot de passe sécurisé.

3 - FRONTEND

3.1 Cross-Site Scripting (XSS)

Contexte : Une attaque XSS (Cross-Site Scripting) est une technique où un attaquant insère du code malveillant (généralement du JavaScript) dans une application web accessible par d'autres utilisateurs. Ce code est alors exécuté côté client, exploitant la confiance que les utilisateurs ont envers le site pour voler des informations sensibles, modifier le contenu affiché, ou prendre le contrôle de sessions utilisateur.

Solution :

Afin de se prémunir des attaques XSS il convient d'appliquer les mesures suivantes:

- L'utilisation d'éléments javascript et CSS inline est à proscrire. La structure doit être clairement établie en séparant HTML, JAVASCRIPT et CSS.
- Toute modification doit être réalisée via l'API DOM.
- Mise en œuvre de Content Security Policy (CSP). CSP permet de contrôler l'origine des ressources accessibles par le navigateur en instaurant une liste blanche. Aussi, une ressource déclarée mais non validée par le CSP se verra bloquée ou ignorée.
- Une attention particulière sera apportée aux formulaires et à toute autre situation ou la saisie d'informations est proposée à l'utilisateur (validation des entrées, échappement des caractères HTML, sanitisation)
- Vérifier l'intégrité des ressources exploitées qu'elles soient d'origine extérieure ou propres au domaine.

3.2 Cross-Site Request Forgery (CSRF)

Contexte : Le CSRF est une attaque où un pirate incite un utilisateur connecté à un site web à exécuter des actions non voulues, comme changer un mot de passe ou effectuer un paiement. Le site traite ces actions comme légitimes parce que l'utilisateur est déjà authentifié, mais l'utilisateur ne les a pas initiées volontairement.

Solution :

Afin de se prémunir des attaques de type CSRF il convient d'appliquer les mesures suivantes:

- Jetons CSRF (CSRF Tokens) : Inclure un jeton unique et aléatoire dans chaque formulaire ou requête sensible. Ce jeton est vérifié par le serveur pour s'assurer que la requête provient de l'utilisateur légitime.
- Vérification du référent HTTP (HTTP Referer Header) : Vérifier que les requêtes proviennent du même domaine que celui du site web, en validant le header `Referer` ou `Origin` de la requête.
- Expiration des sessions : Limiter la durée des sessions et forcer une reconnexion après un certain temps d'inactivité.
- Limiter le transit des cookies aux flux sécurisés
- Définir une stratégie stricte d'envoi des cookies de session en cross-site. Pour un cookie de session, l'attribut samesite doit être défini.

3.3 Clickjacking

Contexte: Le clickjacking est une attaque qui consiste à tromper un utilisateur en lui faisant cliquer sur un élément invisible ou déguisé dans une page web, souvent avec des conséquences nuisibles (par exemple, validation d'un formulaire ou exécution d'une commande sans son consentement). Pour s'en défendre, plusieurs mesures peuvent être mises en place.

Solution : Afin de se prémunir des attaques de type clickjacking il convient d'appliquer les mesures suivantes:

- Mise en place d'une directive CSP interdisant d'incorporer cette application dans un autre site au moyen d'une iframe autre que notre propre domaine (Content-Security-Policy: frame-ancestors 'self')
- Le précédent point sera complété par la directive similaire X-Frame-Options: sameorigin, pour des raisons de compatibilité avec certains navigateurs. Cela contribue à augmenter la défense en profondeur.

3.4 Web Storage, IndexedDB et Cookies

Il convient d'observer quelques règles lors de l'utilisation du stockage local côté navigateur. Un premier élément de sécurité instauré par défaut par le navigateur est la politique de même origine (SoP). A cet égard les précautions suivantes seront mises en place:

- Proscrire le stockage d'informations sensibles dans les bases de données locales et dans les bases de données IndexedDB.
- L'utilisation de l'API Web Sql Database étant obsolète, son usage se voit aussi interdit.

Les cookies sont une solution de stockage temporaire et au volume réduit disposant d'une durée de vie déterminée. Là aussi le stockage de données sensible est à éviter, néanmoins les propriétés de configuration des cookies font de lui un bon réceptacle pour les jetons de sessions sous réserve de paramétrer ces derniers correctement.

4.5 XHR, CORS et Fetch

Par défaut, les navigateurs appliquent une politique de même origine (Same-Origin Policy) pour des raisons de sécurité. Cela signifie qu'un script exécuté sur une page web d'une origine donnée ne peut pas accéder aux ressources d'une autre origine. CORS permet de surmonter cette restriction en définissant des règles spécifiques pour autoriser certaines requêtes inter-domaines. Certaines mesures sont à respecter pour limiter les risques:

- Appliquer une limite sur les origines autorisées. Il est vital d'être précis dans la définition des domaines autorisés. L'utilisation de de l'astérisque pour autoriser toutes les sources est à proscrire.

- Appliquer une limite sur les méthodes HTTP en ne conservant que les méthodes nécessaires (GET, POST, PUT).
- Contrôler les en-têtes autorisés. Définir avec précision les en-têtes autorisées dans les requêtes.
- Configurer les requêtes preflight pour prendre en charge les requêtes options afin de permettre au navigateur de vérifier les autorisations avant d'effectuer la requête principale.

XmlHttpRequest et fetch sont deux solutions pour effectuer des requêtes api. Pour la clarté de sa syntaxe, son meilleur support de CORS et sa simplicité, fetch est à privilégier dans la plus grande majorité des cas. XHR peut néanmoins être utilisé de manière raisonnée pour certains usages spécifiques ou pour des raisons de compatibilité avec les navigateurs plus anciens. Les mesures apportées à l'utilisation des apis sont les suivantes :

- Il est impératif de s’assurer de l’utilisation d’une connexion sécurisée (https://) pour contrer les attaques de type Man in the Middle.
- S’assurer que le contenu de la réponse soit sous un format non exécutable par le client (JSON ou XML).
- Choisir la méthode http adéquate:
    - Limiter au maximum l’emploi de la méthode GET, et uniquement pour la récupération de données publiques non sensibles sans changer l’état des données.
    - Utiliser la méthode POST pour limiter le risque de données.
    - Utiliser la méthode PUT pour limiter le risque CSRF.
    - Ajouter une une configuration CSP et un contrôle anti CSRF pour contrôler les risques

4 - API

L'api fait le lien entre le client et la base de données, de ce fait, de nombreux facteurs sont à prendre en compte concernant la sécurité de cette dernière. Comme présenté dans le paragraphe 2.4 des précautions sont prises au niveau de la transmission des données, et dans le 2.7 sur la politique des mots de passe. Afin de renforcer les connexions entre client et API les jetons de connexions JWT seront employés avec des durées de vie courtes afin de limiter l'impact d'un éventuel vol de token.

4.1 Server-Side Request Forgery (SSRF)

Contexte: Le SSRF est une faille qui permet à un attaquant de tromper un serveur en le forçant à faire des requêtes HTTP vers des ressources internes ou externes, sans qu'il soit directement impliqué.

Solutions :

- Validation et nettoyage des données (sanitisation): Mise en œuvre d'une politique stricte de vérification des entrées utilisateurs afin de vérifier qu'elles ne comportent aucune entrée visant à porter atteinte à l'intégrité de l'application.
- Liste blanche de domaine: Pour éviter l'accès au ressources internes du serveur, une liste blanche de noms de domaines sera utilisée pour bloquer par exemple les adresses IP locales (localhost).

4.2 XXE et Configuration de sécurité

Contexte: Le XXE (XML External Entity), ou injection d'entités externes XML, est une vulnérabilité de sécurité qui se produit lorsqu'un parseur XML non sécurisé traite des données XML fournies par un utilisateur. Cette faille permet à un attaquant d'injecter des entités externes XML malveillantes, pouvant entraîner des accès non autorisés aux fichiers du serveur, une exécution de code à distance, des attaques par déni de service (DoS), et même des attaques sur des services internes via des requêtes HTTP. Dans le rapport de l'OWASP sur les failles de sécurité les plus communes, ce type de faille est désormais classé dans la catégorie "Security Misconfiguration "

Solutions:

- Désactiver les entités externes dans les parseurs XML: Désactiver au si possible cette fonctionnalité reste le moyen le plus sûr de se

prémunir contre cette vulnérabilité.

- Validation et nettoyage des entrées utilisateurs: Un contrôle strict des entrées utilisateur permet d'éviter les entités XML externes malveillantes.
- Privilégier le JSON.
- Limiter les permissions du serveur(moindre privilège): En cas d'attaque XXE réussie, si le serveur n'a que des privilèges minimaux les conséquences seront moindre.
- Restreindre le nombre de features installées ou activées:(ports non nécessaires, services, pages, comptes)
- Supprimer les comptes par défaut.
- Gérer adéquatement les messages d'erreurs: Les messages d'erreurs ne doivent pas donner d'indices sur les dysfonctionnements du service.
- Configuration des options du serveur et du framework: Une attention particulière doit être portée aux options de configuration du serveur et du framework, ainsi que des dépendances installées.

5 - BASES DE DONNÉES

5.1 Injections SQL(SQLi)

Contexte : Une vulnérabilité où un assaillant injecte du code malveillant dans le but de compromettre l'intégrité de la base de données.

Solutions :

- *Utilisation d'un ORM :* Du fait de sa propension à transformer des objets en requêtes SQL, Object-Relational Mapping est un bon candidat pour ajouter une couche de protection contre les injections SQL.
- *Mise en place de requêtes préparées:* Dans le cas où les restrictions apportées par les ORM ne permettent pas de communiquer avec la base de données de la manière souhaitée, les prepared statements doivent être préférés à toute autre méthode. Les concaténations sont à proscrire impérativement.
- *Principe du moindre privilège :* En ne donnant que les permissions adéquates à chaque type d'utilisateurs, le risque de détérioration est amoindri (un élève n'a pas besoin d'écrire dans la table des cours par exemple).
- *Validation et nettoyage des entrées :* Mise en place d'un système de regex qui permet de valider les entrées (le format d'une adresse e-mail), de limiter la taille des entrées où de supprimer des caractères potentiellement dangereux.
- *Journalisation des activités :* Mise en place d'un journal retraçant l'historique des requêtes et surveillant les éventuelles tentatives d'injection SQL.
- *Web Application Firewall :* Mise en place d'un pare-feu applicatif permettant d'analyser le contenu des requêtes et tenter de repérer des motifs suspects. En outre il permet, via des règles prédéfinies d'identifier des requêtes suspectes en se souciant de certains mots-clés spécifiques (exemple : SELECT, UNION, INSERT) où s'adapter à une logique spécifique via des règles personnalisées.

5.2 Attaques par déni de services (DoS)

Contexte : Une attaque visant à saturer en multipliant les requêtes vers un serveur le rendant indisponible pour les utilisateurs.

Solutions :

- *Rate limiting :* Mise en place d'une limitation du nombre de requêtes par utilisateur sur une période donnée visant à réduire le risque d'épuisement des ressources serveur.
- *Surveillance :* Mise en place d'un outil permettant d'analyser le trafic afin de détecter les pics qui pourraient suggérer une attaque DoS.
- *Réplication et répartition de charge :* Mise en place de plusieurs serveurs pour la base de données permettant de répartir le trafic et limiter le risque de surcharge d'un serveur.
- *Caching :* Mise en place d'un système de cache de requêtes pour stocker temporairement les requêtes fréquemment utilisées afin de réduire le nombre d'appel à la base de données.
- *Web Application Firewall :* Mise en place d'un pare-feu qui va analyser le trafic en amont et bloquer les requêtes suspectes avant qu'elles n'atteignent la base de données. Les pare-feux offrent un système de liste noire qui permet de bloquer les adresses IP malveillantes connues.

5.3 Élévation de privilèges

Contexte : Dans le cas où un assaillant arrive à bénéficier de privilèges plus élevés qu'il ne devrait avoir, la sécurité de la base de données serait hautement compromise.

Solutions :

- *Principe du moindre privilège :* En ne donnant que les privilèges nécessaires à l'utilisation prévue pour chaque groupe d'utilisateurs, le risque d'élévation de privilège se voit amoindri.
- *Sécurité renforcée sur les comptes dits « sensibles » :* Limite du nombre de comptes à haut niveau de privilèges et mise en place de mesures plus draconiennes sur la sécurité des comptes avec le plus de privilèges, notamment dans le cas du compte administrateur. Une attention particulière sera accordée à ce dernier (longueur minimum plus longue,

sessions d’authentification plus courtes, mot de passe a durée de vie limitée)

5.4 Sauvegarde des données

Afin de préserver les données des utilisateurs en cas de défaillance matérielle ou logicielle, ou de les protéger contre des attaques de type ransomware visant à rendre les données inutilisable, il est nécessaire d'adopter une stratégie de sauvegarde de la base de données journalière, a une heure ou le pic d'utilisateur est au plus bas afin de ne causer aucune interruption de service. Toutefois il est nécessaire de mettre conjointement en œuvre une stratégie d'effacement des données rétroactive pour se conformer aux obligations légales liées aux concepts de rétention d'informations et de droit à l'oubli.

5.5 Protection des données

Afin de se conformer aux obligations légales imposées par la CNIL, un chiffrement des données sensibles sera mis en place. En outre, on imposera pour ces dernières un UUID comme identifiant. De par leur nature universellement unique, les UUID participent à la sécurité en ne révélant rien de la logique interne (contrairement à l'auto incrémentation).

6 - RGPD

La collecte et le traitement de données personnelles sont réglementés par la RGPD, de fait une attention particulière sera apportée sur ce sujet:

- En commençant par demander le consentement du visiteur pour la collecte et l'utilisation de ses données personnelles.
- En lui exposant de manière claire et transparente comment sont traitées les informations et dans quel but via les mentions légales.
- En ne demandant de renseigner que les données utiles.
- Par ailleurs les données ne seront pas stockées et traitées de la même manière, au vu du caractère sensible de certaines données la CNIL impose des restrictions plus importantes pour certaines données.
- Garder à l'esprit que les visiteurs ayant partagé leurs données personnelles doivent garder le contrôle sur ces dernières. Dans ce cadre trois grands principes seront appliqués:
    - Droit de consultation : les utilisateurs doivent être en mesure de consulter librement à tout moment leurs données personnelles.
    - Droit de rectification : les utilisateurs doivent être en mesure de pouvoir modifier librement les données qu'ils ont partagées.
    - Droit à l'oubli : les utilisateurs ont le droit de pouvoir supprimer de manière complète et définitive les données qu'ils ont communiquées.
- Si le système d'information venait à être compromis, les utilisateurs concernés doivent être avertis sous soixante-douze si la violation représente un risque pour leurs droits et libertés.

Les règles de sécurité proposées tout au long de ce document ont pour vocation d'apporter une sécurité maximale de l'intégrité et de la confidentialité de toutes les données stockées (Privacy by Design) . Cette section a pour but de clarifier les points nécessitant une attention particulière du fait de la réglementation RGPD.

ANNEXE

Tableau RBAC

	Visiteur	Apprenants	Formateurs	Administrateur
Créer un compte (1)	☑			☑
Modifier son propre compte		☑	☑	☑
Modifier n'importe quel compte				☑
Supprimer son propre compte			☑	
Supprimer n'importe quel compte (2)			☑	☑
Créer une formation			☑	
Modifier une formation personnelle (3)			☑	
Modifier n'importe quelle formation (4)				☑
Supprimer une formation personnelle (3)			☑	
Supprimer n'importe quelle formation (4)				☑
Consulter une formation (5)	☑	☑	☑	☑
S'inscrire à une formation			☑	☑
Créer un module			☑	☑
Modifier un module personnel (3)			☑	☑
Modifier n'importe quel module (4)				☑

Supprimer un module personnel [3]			☑	
Supprimer n'importe quel module [4]				☑
Valider ou invalider la réussite d'un module personnel [3]			☑	
Valider ou invalider la réussite de n'importe quel module [4]				☑
Consulter un module [5]	☑	☑	☑	☑
Créer un cours			☑	☑
Modifier un cours personnel [3]			☑	
Modifier n'importe quel cours [4]				☑
Supprimer un cours personnel [3]			☑	
Supprimer n'importe quel cours [4]				☑
Valider ou invalider la réussite d'un cours personnel [3]			☑	
Valider ou invalider la réussite de n'importe quel cours [4]				☑
Consulter un cours [5]	☑	☑	☑	☑
Suivre son avancement personnel		☑		
Suivre l'avancement des apprenants rattachés à ses cours			☑	
Suivre l'avancement de n'importe qui				☑

Communication entre personnes qui suivent la même formation/cours		☑		
Communication bilatérale entre le formateur et l'apprenant inscrit à l'un de ses cours		☑	☑	
Communication avec tout le monde				☑
Consultation de l'historique des messages de n'importe qui (4)				☑

(1) L'administrateur ne jouit pas de la possibilité de créer d'autres comptes Administrateur.

(2) L'administrateur ne pourra pas supprimer son propre compte Administrateur.

(3) Le terme « personnel » est utilisé pour définir les ressources créées par le compte qui y est associé.

(4) Droits octroyés dans le cadre de la lutte contre l'utilisation abusive du service, incluant sans s'y limiter aux règlements des litiges entre apprenants et formateurs ou à la modération en cas de propos violant la charte d'utilisation du service.

(5) Des restrictions peuvent s'appliquer sur l'affichage des ressources pour les visiteurs ou les utilisateurs non-inscrits à une formation.