Sécurité du développement

I. Les différentes failles

A. XSS

Une attaque XSS (Cross-Site Scripting) consiste en l'injection de données dans une page web telles que du code JavaScript ou HTML, afin de déclencher un événement tel qu'une récupération de données.

B. CSRF

Une attaque CSRF (Cross-Site Request Forgery) consiste en l'utilisation d'un site malveillant tiers permettant de forcer un utilisateur authentifié sur un autre site légitime à effectuer des actions non désirées.

C. SSRF

Une attaque SSRF (Server-Side Request Forgery) consiste en une attaque CSRF exécutée au niveau du serveur.

D. SQLi

Une attaque SQLi (SQL injection) consiste en l'injection de code SQL dans une page web où les données entrées sont communicantes avec la base de données. Le but est de casser la requête prévue par le développeur afin d'en créer une nouvelle afin de, par exemple, récupérer des données ou accéder à une partie sensible du site.

E. LFI/RFI

Une attaque LFI/RFI (Local/Remote File Inclusion) consiste en l'exploitation d'un protocole de transfert de fichier afin de transférer des fichiers non prévus par l'application, et de provoquer des erreurs côté serveur ou d'exécuter un code distant par exemple.

F. XXE

Une attaque XXE (XML External Entity) consiste en l'exploitation de l'interpréteur XML afin de charger des données externes et d'exécuter par exemple du code malveillant.

II. Prévention

Faille	Prévention
XSS	❖ Échapper les caractères HTML, CSS et Javascript par exemple grâce aux fonctions php htmlspecialchars() et htmlentities(). Des frameworks comme Angular ou React incluent nativement cette fonctionnalité. ❖ CSP (Content Security Policy) est une norme utilisée pour empêcher les attaques XSS et les injections de code. C'est un en-tête à intégrer aux requêtes HTTP.
CSRF	❖ L'utilisation de la politique SameSite pour les cookies permet d'établir qu'un cookie ne peut être utilisé uniquement par le domaine de l'application.
SSRF	❖ L'utilisation de firewalls à des endroits précis est la pratique la plus courante pour empêcher ce type d'attaques. Cela consiste à définir une whitelist de routes ayant l'autorisation de communiquer avec le serveur. ❖ Une bonne pratique est également ce que l'on appelle le Network Segregation, qui consiste en la segmentation du réseau en sous-réseau ou fragments de réseau, ce qui permet d'interdire les requêtes adressées directement au niveau du réseau et de limiter les dommages en cas d'attaque.
SQLi	❖ Toutes les requêtes SQL doivent utiliser le système de requêtes préparées, permettant d'effectuer la requête indépendamment des arguments.

LFI/RFI	❖ Il convient de définir la configuration des paramètres PHP suivants : ➢ allow_url_open (off) ➢ allow_url_include (off) ➢ open_basedir (défini) Ceci aura pour but d'écarter toute attaque portant sur les fonctions include(), include_once(), require() et require_once() de PHP en lançant un message d'erreur le cas échéant plutôt qu'en traitant le fichier distant.
XXE	❖ La méthode la plus sûre et la plus simple est de désactiver complètement les DTD (Document Type Declaration), c'est à dire les fichiers ou parties de fichier XML.

III. Tests

// Définir les zones à risque et les fonctions à tester