Sécurité des Applications Web

Introduction

Mehdi BENZINE
mehdi.benzine@univ-setif.dz

Département d'Informatique
Faculté des Sciences
Université FERHAT ABBAS Sétif I

2019/2020

Programme I

1. Introduction à la sécurité sur le web
    1. Risques et enjeux
    2. Concepts de sécurité
    3. Sécurité au niveau physique
2. Sécurité au niveau du système d'exploitation
3. Intégrité de l'application web
    1. Vulnérabilités des pages HTML
    2. Intégrité des scripts PHP
    3. Risques causés par les données saisies par les utilisateurs
    4. Téléchargement de fichiers vers le serveur (upload)
    5. Risques liés aux cookies et sessions
4. Sécurité du SGBD
    1. Injection de SQL
    2. Accès au serveur
    3. Accès secondaires
5. Chiffrement et signature des données

Programme II

1. Systèmes de chiffrement symétrique/asymétrique
2. Signature numérique
3. Certificats authentifiés
4. Infrastructure à clé publique (PKI)

6. Sécurité sur le réseau
   1. Vulnérabilités du réseau
   2. Sécurisation d'un ensemble de machines (segmentation en sous-réseaux, filtrage ...)
   3. Chiffrement des communications et authentification des tiers (SSL/TLS, tunneling SSH, IPSec, VPN)

7. Identité numérique

Cours et évaluation

- Cours: 1h30/semaine
- TP: 1h30/semaine
- Évaluation
  - évaluation en TP = tests en TP et/ou interrogations écrites + travail individuel + assiduité
  - note finale = 40% note de contrôle continu + 60% note d'examen
- Contact
  - Bureau: 01
  - Courriel: mehdi.benzine@univ-setif.dz
  - Page web du cours: http://sawinfo.e-monsite.com

Bibliographie

- Sécurité PHP5 et MySQL, Damien Seguy et Philippe Gamache, Eyrolles, 2007.
- Pro PHP Security: From Application Security Principles to the Implementation of XSS Defenses 2^{nd} edition, Chris Snyder, Thomas Myer and Michael Southwell, Apress, 2010.
- Sécurité Informatique Principes et méthode, Laurent Bloch et Christophe Wolfhugel, Eyrolles, 2007.
- Tableaux de bord de la sécurité réseau $2^{ème}$ édition, Cédric Llorens, Laurent Levier et Denis Valois, Eyrolles, 2006.
- Web Security Testing Cookbook, Paco Hope and Ben Walther, O'REILLY, 2008.

Introduction

Les applications web peuvent être détournées de leur utilisation première pour plusieurs raisons:

- Enjeux économiques considérables (e-commerce, e-banking, publicité en ligne ...)
- Grand nombre d'utilisateurs non sensibilisés aux problématiques de sécurité
- Nombre d'applications web en constante augmentation
- Plusieurs langages de programmation (HTML, JavaScript, PHP, Java, Ruby, SQL...)
- Plusieurs plateformes matérielles et logiciels
- Absence de séparation des données et des instructions (représentation textuelle)

Premières notions de sécurité

Parmi les notions fondamentales à connaître pour pouvoir appréhender sécurité:

- Vulnérabilité: faiblesse de l'application qui peut être exploitée pour causer un dommage à l'application web (menace).
- Menace: existence d'une possibilité d'exploitation délibérée ou involontaire d'un vulnérabilité causant un dommage à l'application.
- Exploit: exploitation d'une vulnérabilité pour causer un dommage à l'application web.
- Contre mesure: toute action empêchant l'exploitation d'une vulnérabilité pour causer un dommage à l'application web.

Premières notions de sécurité (suite)

- Périmètre de sécurité: le périmètre de sécurité est l'ensemble des ressources nécessaires au fonctionnement de l'application web et qui se trouvent sous la responsabilité de l'entité en charge d'en assurer la sécurité.
- Risque: quantification de la possibilité de subir des dommages. Le risque peut être calculé et considéré comme élevé, moyen ou faible.
  $$\text{Risque} = \frac{menace\ x\ vulnerabilite\ x\ dommage}{contre\ \ mesures}$$

Sécurité d'une application web

Sécuriser une application web consiste réduire les risques sur les éléments suivants:

- La confidentialité des données
- L'intégrité des données
- La disponibilité de l'application
- La non-répudiation
- L'authentification

Confidentialité

Les données accessibles à travers l'application ne doivent l'être qu'aux seuls utilisateurs autorisés.

La confidentialité concerne les données stockées dans la base de données de l'application ainsi que les données en cours de transit sur le réseau (entre serveur web et serveur BD ou entre serveur web et navigateur).

Plusieurs techniques permettent de s'assurer de la confidentialité des données: contrôle d'accès, cryptographie.

Intégrité

Les données gérées par l'application ne doivent être modifiables que par les utilisateurs habilités à faire cela.

Comme dans le cas de la confidentialité des données, l'intégrité des données s'applique aux données stockées ainsi qu'aux données en transit.

Plusieurs techniques permettent de s'assurer de l'intégrité des données: contrôle d'accès, hachage cryptographique.

Disponibilité

L'interruption du fonctionnement de l'application, volontaire ou accidentelle, pendant une période de temps peut avoir des conséquences financière importantes, des conséquences en terme d'image...

Ces interruptions doivent être évitées autant que possible et leur durée réduite au minimum.

Authentification

Authentifier un utilisateur consiste pouvoir s'assurer de son identité réelle.

L'authentification peut être réalisée par différents moyens tel que l'utilisation d'un mot de passe, une information biométrique (empreinte digitale, iris de l'œil, forme du visage...), une clé cryptographique privée, un token matériel d'authentification...

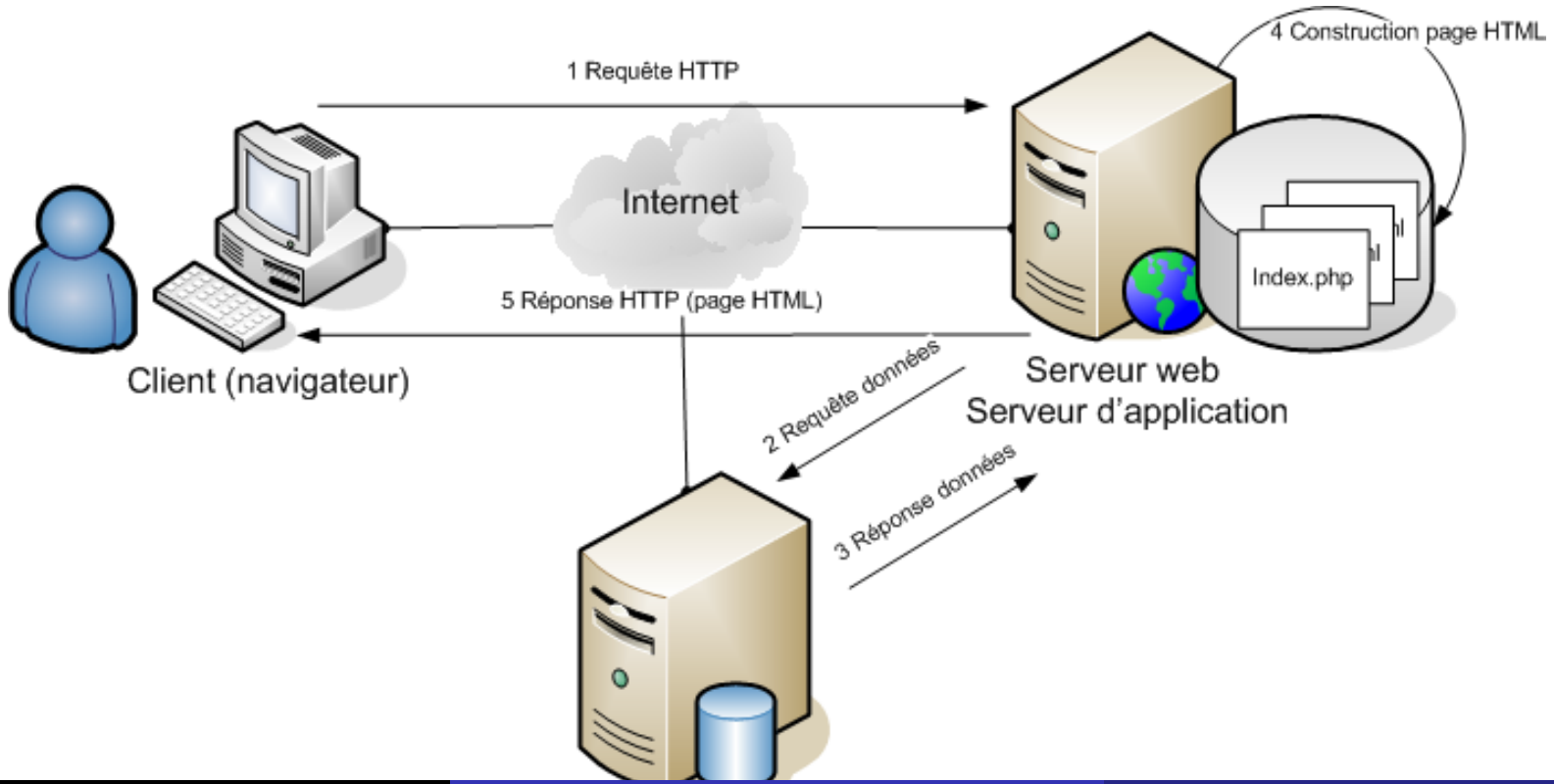
Non-répudiation

La non-répudiation consiste à conserver des preuves fiables sur l'identité des acteurs ayant effectuer des actions de manière à ce qu'ils ne puissent pas nier avoir été à l'origine de ces actions.

La non-répudiation permet de données aux transactions réalisées à travers une application web une validité contractuelle et légale.

Caractéristiques des applications web

- Applications distribuées
- Diversité des plateformes matérielles et logicielles utilisées
- Diversité des langages de programmation utilisée (HTML, CSS, JavaScript, PHP, Java, Ruby, SQL...)
- Utilisateurs aux profils divers et non sensibilisés à la problématique de la sécurité



Concepts de sécurité

La sécurité d'une application web doit être une préoccupation constante tout au long de son cycle de vie.

Elle doit être prise en compte au moment de
(1) la conception,
(2) du développement
(3) puis de manière quotidienne par un suivi du fonctionnement après la mise en production.

Sécurité à la conception I

- La phase de conception d'une application est capitale pour en garantir la sécurité.
- Les bonnes pratiques et les choix judicieux faits lors de cette étape se révèleront bénéfiques en terme de sécurité lors de la mise en production.
- Il est difficile de corriger, au moment du développement, ou après la mise en production, des problèmes de sécurité qui découlent d'une mauvaise conception.
- Penser la sécurité au moment de la conception est toujours profitable.
- Pour faire cela, voici quelques règles simples à adopter au moment de la conception:
  - Faire preuve de bon sens: Ne faire que ce qui est utile et nécessaire. Éviter l'envoi de données sensibles sur le réseau quand cela n'est pas nécessaire ...

Sécurité à la conception II

- Faire simple: une conception et un code simple sont plus faciles à analyser et à comprendre qu'une conception et un code complexe.
- Sécuriser par l'obscurité: cacher toutes les informations possibles sur l'application web. Plus un attaquant connait d'information sur l'application, plus il a de chance de trouver une vulnérabilité. Cacher les messages d'erreur, les warnings, l'URL des fichiers, les outils/framewoks/technologies utilisées ...
- Défendre en profondeur: multiplier les barrières pour empêcher un attaquant d'atteindre son but. S'il franchit une barrière , une autre barrière est là pour l'arrêter. Faire des tests de sécurité à plusieurs niveaux de l'application (navigateur/JavaScript, serveur web/PHP, SGBD/contrainte d'intégrité) même s'ils sont redondants.

Sécurité pendant le développement

La sécurité pendant le développement relève de la responsabilité des développeurs.
Cela consiste à implémenter la spécification issue de la conception tout en veillant à produire un code n'offrant pas de vulnérabilités.
Pour cela, voici quelques règles de sécurité à respecter lors du développement d'une application web:

- Filtrer les données entrantes: les données entrées par l'utilisateur doivent être contrôlées pour vérifier qu'elles sont conforme à ce que l'application attend (type, format, taille ...) et vérifier qu'elles n'ont pas de sens particulier pour le langage utilisé (<, >, ...)
- Se méfier du jeu de caractères de l'utilisateur: le jeu de caractères que l'utilisateur prétend utiliser peut servir à cacher le texte réellement envoyé. Envoyer du code HTML en UTF-8 et prétendre que c'est de l'ASCII.

Sécurité pendant le développement (suite)

- Suivre les données: les données saisies par l'utilisateur doivent être suivies tant qu'elles sont utilisées par l'application. Il faut savoir en permanence pour chaque donnée si elle a été filtrée et vérifiée ou non.
- Protéger les données en sortie: les données en sortie de l'application étant destinées à être utilisées par un autre langage, il faut s'assurer que ces données ne sont pas "toxiques" pour ce langage.
- Auditer le code: le code produit doit être audité par un développeur externe au projet. Le regard neuf d'un expert extérieur peut trouver des vulnérabilités plus facilement qu'un développeur impliqué sur un projet.

La sécurité pendant le développement consiste essentiellement à mettre en place les mécanismes de filtrage des données/fichiers ... fournit par l'utilisateur. Chaque langage de programmation fournit des mécanismes pour faire cela.

Sécurité au quotidien

Même après la mise en production, la sécurité d'une application n'est toujours pas garantie. Il s'agit maintenant de surveiller l'application et de vérifier qu'elle réalise bien les tâches qui sont prévues et rien d'autre. Dans le cas contraire, il faut prévoir les protocoles et les procédures à appliquer pour remédier à cela.

Pour faire, cela voici quelques règles de sécurité à adopter:

- Modérer les contenus publiés par les utilisateurs (si l'application offre un forum, blog ...). Supprimer les messages qui ne sont pas conformes à la loi et/ou la charte d'utilisation du site.
- Analyser régulièrement les fichiers de log: vérifier les fichiers de log de l'application et de tous les outils/serveurs qu'elle utilise. Cela permet de vérifier que l'application fonctionne de manière normale.

Sécurité au quotidien (suite)

- Appliquer régulièrement les mises à jours de sécurité: mettre à jour dès que possibles les bibliothèques, outils, serveurs, systèmes d'exploitation ... utilisés par l'application web.

Sécurité sur 4 niveaux architecturaux

Sécuriser une application web nécessite de sécuriser l'ensemble des composants matériels et logiciels nécessaires au fonctionnement de celle-ci.
La présence d'une vulnérabilité à quelque niveau que cela soit provoque la vulnérabilité de toute l'application.
Pour sécuriser une application web, il est nécessaire de connaître tous les composants matériels et logiciels qui la font fonctionner.
Il y a lieu de distinguer 4 niveaux architecturaux:

Niveau physique (matériel)

Matériel nécessaire au stockage et à l'exécution de l'application web.

- Serveurs (serveur web, serveur d'application, serveur BD)
- Postes clients
- Équipements réseau (routeurs, hubs, switchs, câbles...)

Niveau physique (matériel)(suite)

Vulnérabilités:

- Stockage dans des locaux inadaptés (facilité d'accès aux personnes, mauvaise aération, humidité...)
- Composants matériels de mauvaise qualité
- Alimentation électrique et connexion réseau défaillantes
- ...

Dommages:

- Vol, incendie, dégât des eaux...
- Panne
- Coupure de l'alimentation électrique et/ou de la connexion réseau
- ...

Contre mesures:

- Locaux fermés et surveillés, portes coupe-feu, armoires blindées...
- Composants matériels de bonne qualité, redondance
- Onduleurs, générateurs de courant, connexion réseau fiable...

Niveau système d'exploitation

Les systèmes d'exploitation modernes offrent des mécanismes de sécurité robustes.

Ils permettent de définir des comptes utilisateurs et de leur attribuer des privilèges et autorisations sur le système (droit d'administration, droit de lire ou d'écrire dans un répertoire ou fichier, droit de démarrer ou d'arrêter un service...).

L'utilisation d'un compte nécessite de passer par un processus d'authentification de manière à s'assurer de l'identité de l'utilisateur (identifiant + mot de passe, clé cryptographique privée, données biométriques...).

Niveau système d'exploitation (suite)

Vulnérabilités:

- Comptes sans mot de passe
- Mot de passe trop court, trop simple...
- Gestion laxiste des droits d'utilisateur (utilisateurs disposant de plus de droits que nécessaire)
- Négligence dans l'application des mises à jour de sécurité
- ...

Dommages:

- Usurpation d'identité
- Accès illicite aux données
- Destruction de données
- ...

Contre mesures:

- Imposer l'utilisation de mots de passe robustes (longs, plusieurs types de caractères...)
- Limiter les droits de chaque utilisateurs uniquement à ceux nécessaires pour accomplir ses tâches

Niveau application

La sécurité de l'application web doit être une préoccupation constante tout au long de son cycle de vie.

Vulnérabilités:

- Absence de filtrage des données saisies par l'utilisateur (type, taille, format...)
- Absence de filtrage des fichiers téléchargés vers le serveur (type, taille ...)
- ...

Dommages:

- A voir dans les prochains cours

Contre mesures:

- A voir dans les prochains cours

Niveau réseau

Les applications web étant des applications réparties, les communications réseau sont nécessaires à leur fonctionnement. La sécurité sur le réseau consiste essentiellement à sécuriser les échanges de données et à garantir l'identité des tiers qui participent à la communication.

Vulnérabilités:

- Pas de sécurité native sur Internet.
- Par défaut, les communications sont en clair.
- Facilité d'espionnage sur un réseau local.
- Absence d'authentification des tiers.

Menaces:

- A voir dans les prochains cours

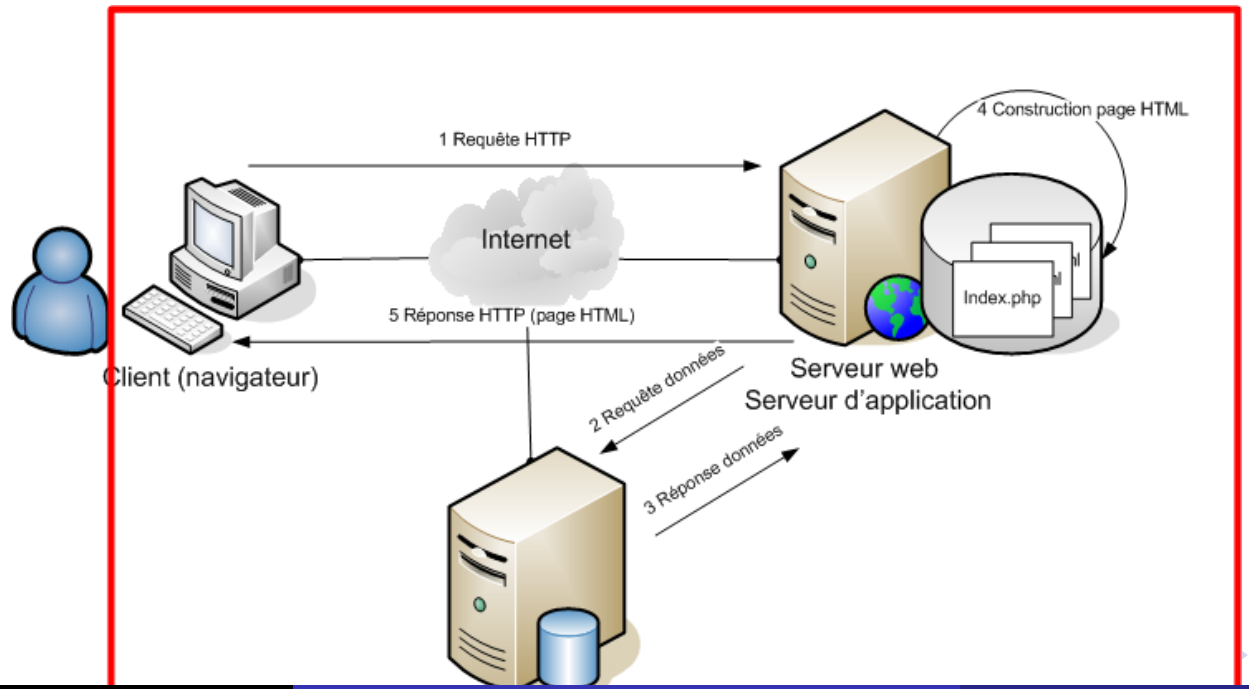
Contre mesures:

- A voir dans les prochains cours

Périmètre de sécurité (1)

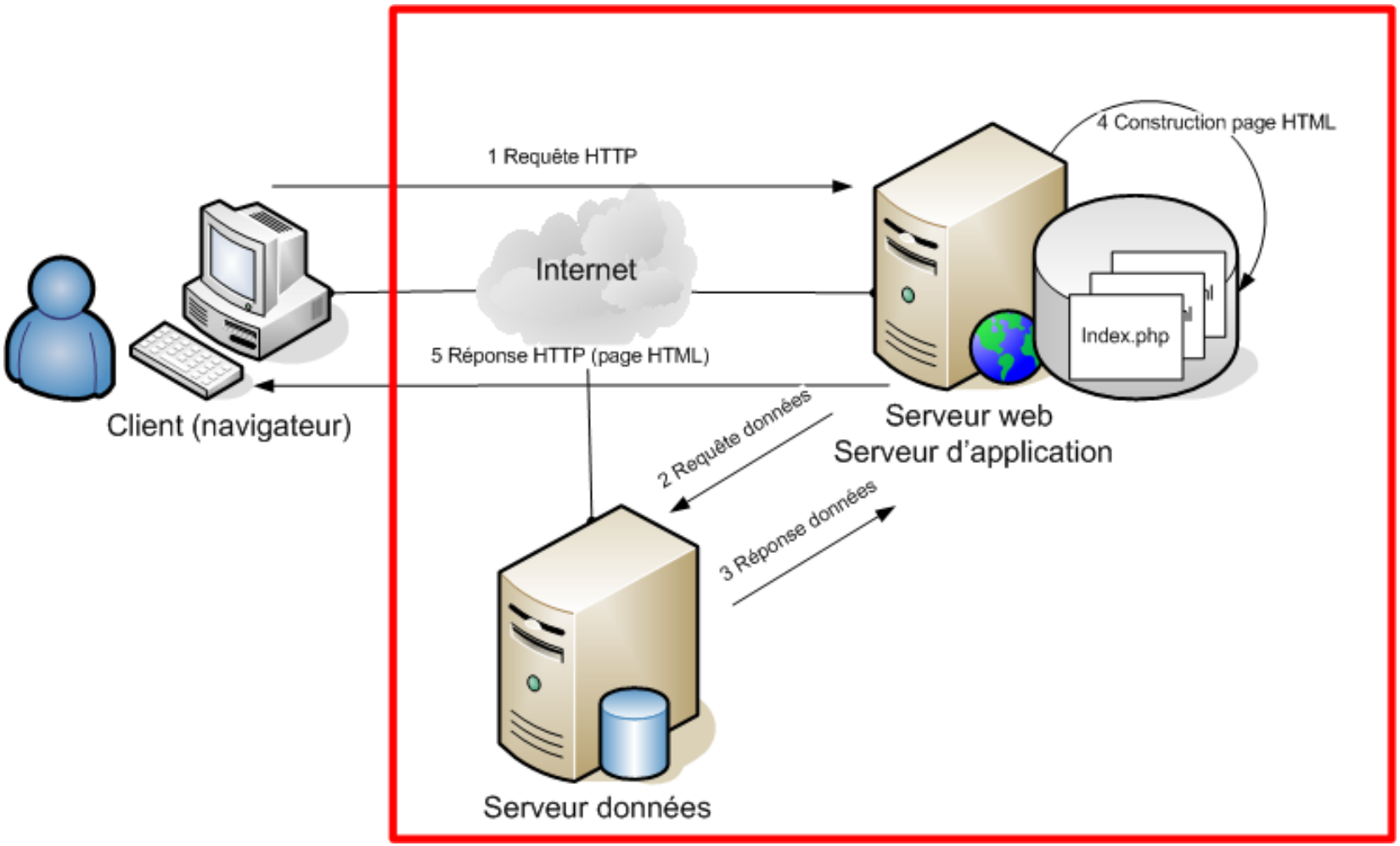
Périmètre de sécurité (rappel): le périmètre de sécurité est l'ensemble des ressources nécessaires au fonctionnement de l'application web et qui se trouvent sous la responsabilité de l'entité en charge d'en assurer la sécurité.

Protéger l'ensemble de l'application web, les infrastructures utilisées et communications.



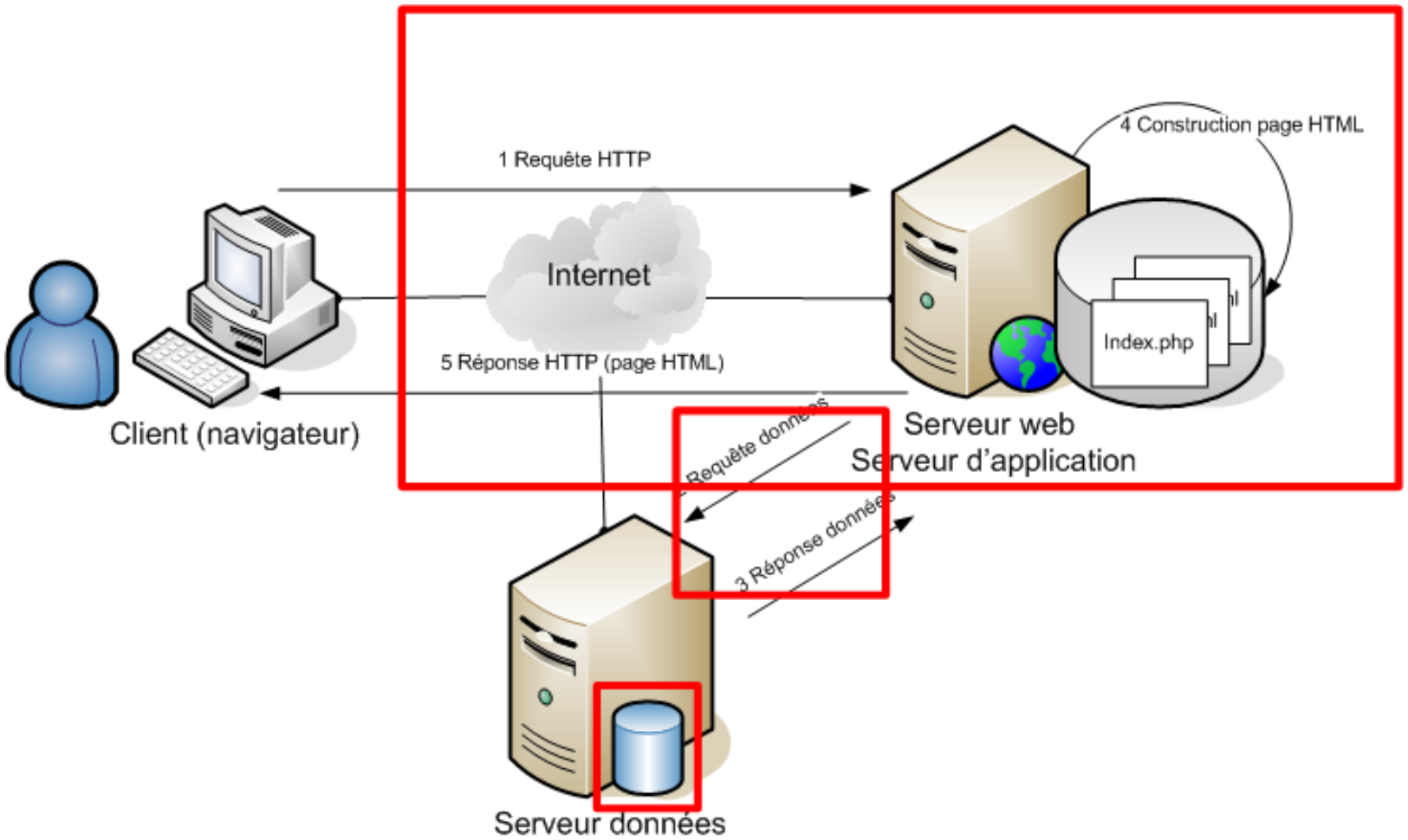
Périmètre de sécurité (2)

Protéger l'application mais pas le poste client. Les utilisateurs de l'application n'appartiennent pas à l'organisation.



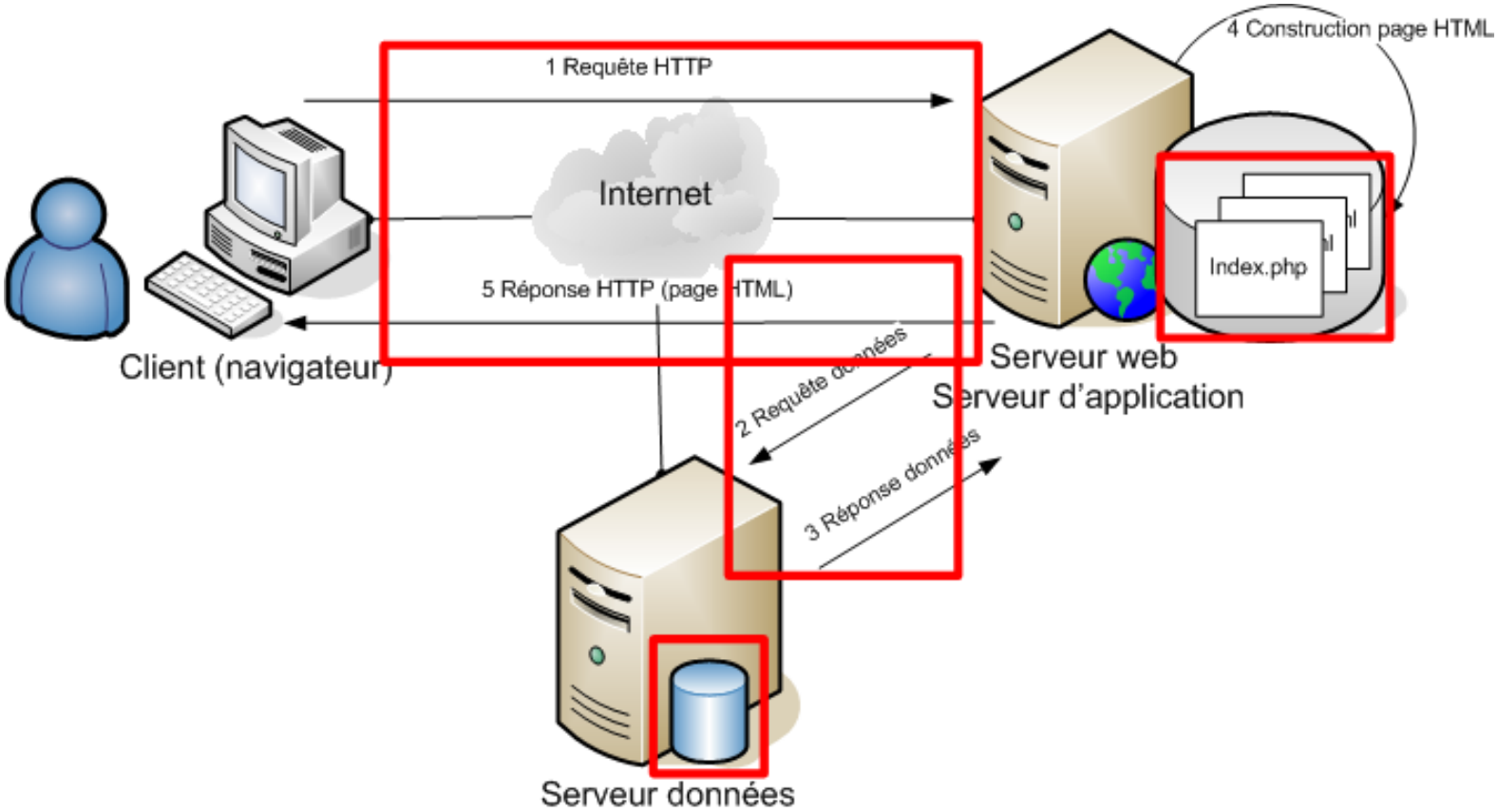
Périmètre de sécurité (3)

Protéger l'application sans la base de données. Les données sont hébergées par un prestataire externe.



Périmètre de sécurité (4)

Protéger l'application et les communications mais pas les infrastructures. L'application et les données sont hébergées par un prestataire externe.



Conclusion

- La sécurité est une notion difficile à appréhender, à mesurer ...
- La sécurité doit être prise en compte tout au long de la vie de l'application web.
- La sécurité doit prendre en considération l'ensemble des composants matériels et logiciels nécessaires au fonctionnement de l'application.
- La sécurité absolue n'existe pas.