TITRE PROFESSIONNEL
DÉVELOPPEUR WEB ET WEB MOBILE

skill&you.

TP DÉVELOPPEUR WEB ET WEB MOBILE
EXTRAITS DE COURS

Voici des extraits de cours de votre future formation de développeur web et web mobile.

Bonne lecture !

skill&you.

EXTRAITS DE COURS
SOMMAIRE

Envie de découvrir les contenus sur lesquels vous allez bâtir vos compétences ?
Vous trouverez ici quelques extraits de cours qui composent votre futur formation.

01

Extrait n°1.

Comprendre les offres
d'hébergements

02

Extrait n°2.

Qu'est-ce que le JavaScript ?

03

Extrait n°3.

Rechercher une réponse
à une difficulté technique

04

Extrait n°4.

Les principales failles de sécurité
des applications web
et leurs parade

Extrait n°1.

COMPRENDRE LES OFFRES D'HÉBERGEMENTS

Dans le monde numérique actuel, la capacité à distinguer et à comprendre les différentes offres d'hébergement est devenue une compétence incontournable.

Comprendre les offres d'hébergement

Dans le monde numérique actuel, la capacité à distinguer et à comprendre les différentes offres d'hébergement est devenue une compétence incontournable. Ces connaissances permettent de naviguer de manière avisée dans un environnement en constante évolution, tout en répondant efficacement aux besoins spécifiques des projets en ligne. L'objectif de ce module est d'établir un cadre pour l'analyse des options d'hébergement disponibles, en soulignant leur importance et leur impact dans le contexte contemporain.

I. Contextualisation : tendances et évolutions récentes

Dans cette section, nous nous pencherons sur les forces motrices actuelles qui influencent le domaine de l'hébergement web. À travers une analyse méthodique, nous explorerons l'émergence du **cloud computing** et la raison pour laquelle il est devenu un choix dominant pour de nombreux acteurs. En outre, nous discuterons de l'importance croissante de la **cybersécurité** lors de la sélection d'un hébergement et nous verrons comment les **préoccupations environnementales** façonnent les offres d'hébergement écoresponsables.

Le but est de fournir une vue d'ensemble bien informée qui servira de fondement pour une compréhension plus approfondie des **spécificités des offres d'hébergement** dans la suite des développements. Cela facilitera la sélection de l'option d'hébergement la plus appropriée, en tenant compte des facteurs clés qui influencent actuellement ce domaine.

Nous entrerons dans les détails de l'émergence du cloud computing, en explorant les raisons de sa domination dans l'industrie de l'hébergement web.

A. Émergence du cloud computing : pourquoi est-il devenu dominant ?

Pour comprendre pourquoi le cloud computing occupe une place prédominante dans l'industrie actuelle, il est essentiel de revenir sur ses origines et d'examiner les facteurs qui ont contribué à son essor.

Fig. 1 Cloud computing © madedee - stock.adobe.com

1. Flexibilité et évolutivité

Le cloud computing offre une **flexibilité** et une **évolutivité** qui sont difficilement égalées par les infrastructures physiques traditionnelles. Les entreprises et les individus peuvent **ajuster leurs ressources** en fonction de leurs **besoins**, souvent en temps réel. Cette flexibilité a permis une **réduction significative des coûts**, car elle évite les investissements initiaux lourds dans l'infrastructure physique.

2. Accès global

Grâce au cloud computing, les services et les données sont **accessibles** de n'importe où dans le monde, à condition d'avoir une connexion internet. Cette disponibilité globale a facilité la **collaboration** et la **communication**, rendant ainsi les opérations **plus fluides et plus efficaces**, surtout dans un contexte où le travail à distance est devenu courant.

3. Mise à jour et maintenance simplifiées

Les fournisseurs de services cloud gèrent la maintenance et les mises à jour, ce qui dispense les entreprises d'allouer des ressources pour ces tâches. Cela garantit non seulement que les systèmes sont **toujours à jour**, mais **réduit également les temps d'arrêt potentiels**.

4. Sécurité renforcée

Bien que la sécurité puisse initialement être perçue comme une faiblesse du cloud, de nombreux fournisseurs ont investi de manière significative dans des mesures de sécurité robustes. Cela inclut des **protocoles avancés de cryptage des données** et des **protections contre les menaces physiques et virtuelles**.

5. Modèles économiques avantageux

Les **modèles de facturation basés sur l'utilisation**, typiques du cloud computing, permettent une gestion financière plus précise et plus efficace. Les entreprises peuvent planifier leurs dépenses de manière plus pointue, en payant uniquement pour les **ressources qu'elles utilisent réellement**.

Pour offrir une perspective plus complète, considérons un tableau comparatif qui met en évidence les avantages du cloud computing par rapport aux solutions traditionnelles.

Tableau n°1 Les avantages du cloud computing par rapport aux solutions traditionnelles

ASPECTS	CLOUD COMPUTING	SOLUTIONS TRADITIONNELLES
Coûts initiaux	Faibles (modèle pay-as-you-go)	Élevés (achat d'infrastructure)
Évolutivité	Élevée (ajustements en temps réel)	Limitée (mise à niveau physique requise)
Maintenance	Prise en charge par le fournisseur	Gérée en interne (coûts et ressources supplémentaires)
Accès	Global (connexion internet requise)	Limité (accès physique ou réseau interne)
Sécurité	Forte (gros investissements des fournisseurs)	Variable (dépend de l'investissement interne)

En combinant ces avantages intrinsèques avec un écosystème technologique en évolution rapide, il est facile de voir pourquoi le cloud computing occupe une **position dominante** dans l'industrie.

Dans la section suivante, nous nous concentrerons sur un autre aspect crucial de l'hébergement web : la cybersécurité et son importance croissante.

B. L'importance croissante de la cybersécurité dans l'hébergement

Dans le contexte actuel, où les activités numériques sont omniprésentes, la cybersécurité dans l'hébergement est devenue un pilier central garantissant la sécurité et la stabilité des opérations en ligne.

Voici quelques aspects cruciaux qui soulignent son importance croissante.

1. Sécurité des données

Dans le secteur de l'hébergement, la **sécurisation des données est primordiale**. Les incidents de sécurité, tels que les violations de données, peuvent avoir des conséquences dévastatrices, non seulement en termes de **pertes financières**, mais aussi de **réputation**. Les mesures comme le **cryptage des données** et les **pare-feu robustes** sont donc essentielles pour protéger les informations sensibles contre les accès non autorisés.

2. Conformité réglementaire

De plus en plus, les **organismes gouvernementaux** établissent des normes strictes en matière de cybersécurité, exigeant que les entreprises mettent en œuvre des **protocoles de sécurité renforcés** pour préserver les données des utilisateurs. La conformité à ces réglementations est non seulement une **obligation légale**, mais elle permet également d'instaurer la **confiance** avec les **clients** et les **partenaires commerciaux**.

3. Protection contre les attaques DDoS

Les attaques **DDoS (*Distributed Denial of Service*)** sont des stratégies couramment utilisées pour perturber le fonctionnement des services hébergés, en les **submergeant de trafic** jusqu'à ce qu'ils deviennent **inaccessibles**. Les fournisseurs d'hébergement doivent être équipés pour détecter et atténuer ces attaques rapidement afin de garantir la continuité du service.

4. Surveillance et réponse aux incidents

Un élément clé de la cybersécurité est la capacité à **surveiller activement** les systèmes pour détecter toute activité suspecte ou non autorisée. Cela est complété par des **plans de réponse aux incidents** bien élaborés qui permettent une **intervention rapide et efficace** en cas de problème de sécurité.

Pour mieux appréhender ces éléments, voici un tableau qui illustre différents composants de la cybersécurité dans l'hébergement et leurs rôles respectifs.

Tableau n°2 Les composants de la cybersécurité dans l'hébergement et leurs rôles

COMPOSANTS	RÔLES	EXEMPLES D'OUTILS/DE MESURES
Cryptage des données	Protège les données contre les accès non autorisés.	Protocoles SSL/TLS
Pare-feu	Bloque les trafics non sécurisés et non autorisés.	Pare-feu matériels et logiciels
Anti-malware	Protège contre les logiciels malveillants.	Solutions antivirus
Gestion des patchs	Maintient les systèmes à jour et sécurisés.	Outils de gestion des patchs
Surveillance de réseau	Détecte les activités suspectes.	Systèmes de détection d'intrusion (IDS)
Plans de réponse aux incidents	Permet une intervention rapide en cas de problème de sécurité.	Équipes d'intervention d'urgence informatique (CERT)

Il est crucial d'aborder la cybersécurité comme un effort continu et multifacette. En alliant des **mesures proactives et réactives**, on peut construire une infrastructure d'hébergement robuste et résiliente face aux menaces de sécurité en constante évolution.

Dans la section suivante, nous aborderons les préoccupations environnementales associées à l'hébergement et les solutions écoresponsables disponibles.

C. Préoccupations environnementales et hébergement écoresponsable

Dans une époque où les préoccupations environnementales sont au premier plan, l'impact des technologies de l'information sur l'environnement est devenu un sujet de discussion significatif. Les **centres de données**, en particulier, sont des consommateurs voraces d'énergie.

Par conséquent, il est impératif de considérer des options d'hébergement qui soient à la fois **performantes et respectueuses de l'environnement**.

Voici quelques axes principaux pour naviguer dans cette tendance importante.

1. Impact environnemental des centres de données

Les centres de données consomment une **grande quantité d'énergie** pour maintenir les serveurs en fonctionnement et pour les **systèmes de refroidissement** nécessaires à leur bon fonctionnement. De plus, ils génèrent une quantité significative de **déchets électroniques**. Comprendre ces impacts est la première étape vers le choix de solutions plus durables.

Fig. 2 Salle de centre de données internet avec serveur © Yanawut Suntornkij - stock.adobe.com

2. Solutions écoresponsables

Heureusement, de nombreuses initiatives cherchent à atténuer l'impact environnemental des centres de données. Voici quelques stratégies courantes employées :

- **énergie renouvelable** : utiliser de l'énergie provenant de sources renouvelables, comme le solaire ou l'éolien, pour alimenter les centres de données ;
- **refroidissement efficace** : mettre en œuvre des méthodes de refroidissement plus efficaces pour réduire la consommation d'énergie ;
- **recyclage des déchets électroniques** : instaurer des programmes de recyclage afin de gérer les déchets électroniques de manière responsable.

Extrait n°2.

QU'EST-CE QUE LE JAVASCRIPT ?

Grâce à une bonne évolution et à un ensemble d'outils, il est devenu relativement facile et agréable à développer des applications web, mobiles et de bureau avec JavaScript.

Qu'est-ce que le JavaScript ?

Mal aimés pendant quelques années, mais indispensables pour tout développement web, le langage et l'écosystème ont su évoluer et devenir l'un des langages de programmation les plus complets et les plus populaires. Grâce à une bonne évolution et à un ensemble d'outils, il est devenu relativement facile et agréable de développer des applications web, mobiles et de bureau avec JavaScript.

I. Introduction

A. Présentation

Fig. 1 Logo JavaScript © Michel Diemer

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et, à ce titre, il est une partie essentielle des applications web. Avec les langages HTML5 et CSS3, JavaScript est au cœur des langages utilisés par les développeurs web. La grande majorité des sites web l'utilise.

JavaScript est un langage web interprété, gratuit et *open source* (le code source est disponible).

B. Le terme « JavaScript »

JavaScript a été créé en 1995 par Brendan Eich. Nommé en interne « Mocha », il a été publié sous le nom de « LiveScript », en septembre 1995, dans le navigateur web Netscape Navigator 2.0 (ancêtre de Mozilla Firefox).

Le 4 décembre 1995, surfant sur la vague de popularité des *applets* Java (qui dynamisent les pages web à partir du 23 mai 1995), il est renommé « JavaScript ». Il s'agit de la contraction de « Java » et de « LiveScript ». Et Microsoft, pour des raisons légales, a contracté cela en « JScript » dans son navigateur Internet Explorer (qui est obsolète à ce jour).

Plusieurs variantes du langage ont coexisté et, pendant des années, il a fallu les gérer. En juin 1997, JavaScript a été standardisé sous le nom d'« ECMAScript » par Ecma International, dans le standard ECMA-2. En 2006 est sortie la librairie jQuery qui facilite l'écriture de code JavaScript et harmonise les différentes variantes.

C. Nom officiel

ECMA : European Computer Manufacturers Association.

ECMAScript est un standard de programmation pour les langages de scripts sur lequel reposent JavaScript ou encore ECMA-262. JavaScript implémente la spécification officielle décrite par ECMA-262 et y ajoute un ensemble d'API qui le rendent utile et puissant.

Référence officielle du JavaScript ECMA-262

D. Quelques caractéristiques de JavaScript

Fig. 2 Validation © Adobe Stock

- Syntaxe de base inspirée du langage C avec des points-virgules à la fin de chaque ligne, des accolades pour définir les blocs d'instruction et les instructions de base.

- Langage de script interprété, directement disponible dans le navigateur web. Pas besoin d'installer quoi que ce soit.

– Langage objet à base de prototype, contrairement à Java ou PHP qui sont des langages à base de classe. Les objets sont définis dynamiquement.

– Format de données JSON (*JavaScript Object Notation*), à la fois simple et complet, qui permet de convertir un objet en texte et du texte en objet de manière très précise. Plus rigoureux que le CSV et beaucoup plus simple que l'XML, il permet de transporter facilement des données entre le navigateur et le serveur distant.

– Le DOM (*Document Object Model*) permet de manipuler du code HTML (ajouter ou retirer du code HTML, ajouter ou retirer des classes et des attributs, valider des formulaires, etc.) très facilement.

– Depuis 2017 (version 8) : « async/await » simplifie grandement l'écriture de code JavaScript.

– Typage dynamique par défaut (depuis 2012, on peut utiliser TypeScript pour plus de rigueur).

II. Bref historique

A. Web statique

Au début du Web, tout se faisait côté serveur.

Fig. 3 Intérieur de la salle des serveurs © Scanrail – Fotolia

À chaque fois qu'il fallait modifier quelque chose dans une page web, le navigateur internet devait interroger le serveur et actualiser toute la page.

B. Première version de JavaScript

Tableau n°1 JavaScript jusqu'en 2009 : les bases du langage

VERSION	NOM OFFICIEL	NOUVEAUTÉS
ES1	ECMAScript 1 (1997)	Première édition
ES2	ECMAScript 1 (1998)	Corrections éditoriales
ES3	ECMAScript 1 (1999)	try/catch, do/while, switch, etc.
ES5	ECMAScript 1 (2009)	JSON, String.trim(), etc.

C. Web dynamique

Pour créer de véritables applications web, JavaScript permet de :

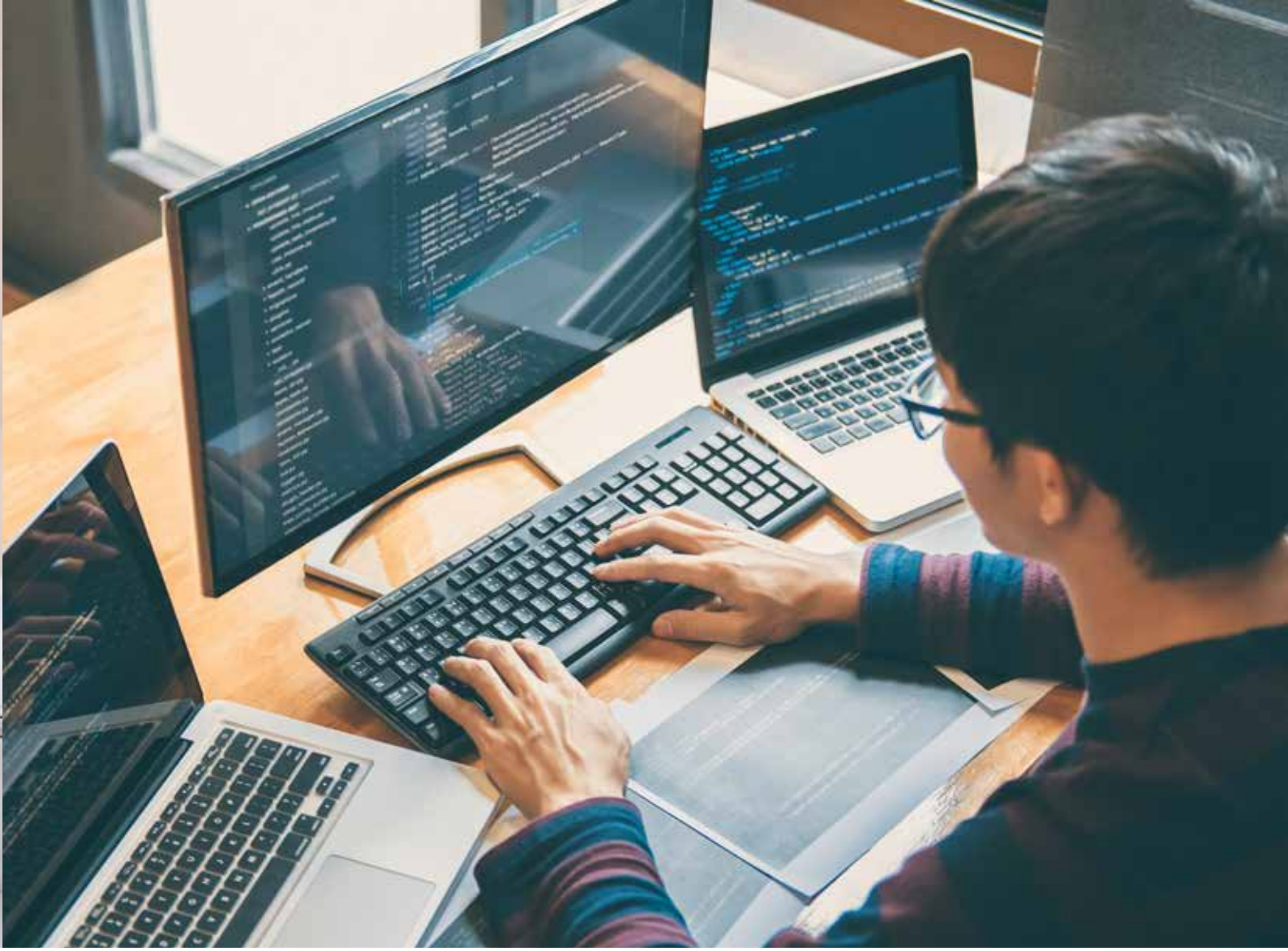
– modifier à la volée le code HTML et donc ajouter dynamiquement du contenu. Plus besoin de recharger le menu ou le *footer* ;

– charger des données progressivement grâce aux requêtes Ajax et les afficher au fur et à mesure qu'elles sont disponibles ;

– répondre à des événements actionnés par l'utilisateur (survol de la souris, appui sur une touche de clavier, etc.), planifiés (vérification toutes les cinq secondes si de nouvelles données sont disponibles), provoqués par le navigateur, etc.

L'utilisation massive de ces possibilités fut une contribution essentielle à l'essor de ce qu'on nomme le « Web 2.0 » et à l'apparition de systèmes « temps réel » comme le sont les réseaux sociaux. Si vous essayez d'imaginer un Facebook où il vous faudrait actualiser sans cesse votre affichage pour avoir les mises à jour des commentaires, des photos ou autres informations, vous conviendrez vite que ce site web n'aurait pas pu avoir le succès qu'il connaît aujourd'hui.

JavaScript permet de dynamiser une page web. Avant HTML5 et CSS3, JavaScript était utilisé pour créer des menus, des effets de texte, des animations et des transitions dans les pages web. Aujourd'hui, il est utilisé côté client, conjointement avec le CSS, pour créer des composants d'interface graphique comme des calendriers, par exemple.



Extrait n°3.

RECHERCHER UNE RÉPONSE À UNE DIFFICULTÉ TECHNIQUE

Lorsque vous êtes confrontés à une difficulté technique de développement, la première étape est de correctement identifier le problème et sa cause.

Sécurité de l'application sous Wordpress

I. Sécurité de l'application sous WordPress

A. Introduction : principes généraux de sécurité

Même si votre site n'a aucune raison d'attirer des actes malveillants, sa sécurité doit être une préoccupation : la grande majorité des tentatives d'attaque sur Internet sont réalisées par des robots parcourant automatiquement le Web à la recherche de la moindre vulnérabilité.

Si un hacker parvient à infiltrer votre site, il pourra le rendre inaccessible, y injecter des publicités indésirables, du code malveillant destiné à infecter l'ordinateur de vos visiteurs (*cross-site scripting*). De tels événements auraient un impact néfaste et durable sur la confiance de votre public ainsi que sur votre indexation auprès des moteurs de recherche.

La sécurité d'un site web
« La sécurité d'un site web » - Mozilla.org
https://developer.mozilla.org/fr/docs/Learn/Server-side/First_steps/Website_security

Les sites web utilisant WordPress sont parmi les plus piratés.

CMS : 90 % des sites web piratés utilisent WordPress
« CMS : 90 % des sites web piratés utilisent WordPress »
https://www.phonandroid.com/cms-90-des-sites-web-pirates-utilisent-wordpress.html

Cela se comprend aisément : WordPress est le CMS (Content Management System, ou système de gestion de contenu) le plus utilisé sur Internet et donc le plus intéressant à cibler lors d'une attaque de masse. De plus, étant très facile à prendre en main, il compte de nombreux utilisateurs amateurs et peu au fait des pratiques de sécurité.

Les attaques contre un site utilisant WordPress (ou tout autre CMS) se regroupent principalement en deux catégories :

- celles exploitant des **failles internes du CMS** ou des **extensions et thèmes installés**. Dès qu'une telle faille est découverte par les équipes de développement, elle fait l'objet d'une correction dans une mise à jour de sécurité : il est donc très important de maintenir son site **constamment à jour** ;

- celles exploitant des failles liées aux **mauvaises pratiques d'administration** d'un site.

Ce cours a pour objectif de vous offrir des bases solides en matière de sécurisation d'un site WordPress, sans que soient nécessaires des notions de programmation ni d'administration de serveur.

B. Lors de l'installation

Si vous installez vous-même votre site WordPress, il est recommandé de le faire via un **accès direct au serveur**, préférable aux installations « one-click » parfois proposées par des hébergeurs.

Assurez-vous que le serveur sur lequel vous effectuez cette installation dispose d'une **version PHP récente** telle que recommandée dans les prérequis WordPress (https://fr.wordpress.org/support/article/requirements/).

Vous risqueriez sinon de ne pas bénéficier des dernières mises à jour de sécurité et d'avoir un site vulnérable.

Pour débuter l'installation, la procédure détaillée et conseillée est disponible dans la documentation officielle WordPress : https://fr.wordpress.org/support/article/how-to-install-wordpress/

Lors de ce processus d'installation, la partie la plus importante concernant la sécurité est le choix de l'**identifiant du compte administrateur** et de son **mot de passe**. Ce compte sera votre principal outil pour accéder à l'interface d'administration WordPress et aura tous les droits (y compris celui de supprimer le site !) : sa sécurité est de ce fait primordiale.

Fig. 1 Écran d'installation de WordPress. Encadré en rouge : choix de l'identifiant administrateur et de son mot de passe. © Richard Legrand

Évitez « admin », qui est l'identifiant administrateur traditionnel sous WordPress et représente donc un risque de sécurité, ou tout autre identifiant générique type « root » ou « administrateur » : utilisez un nom de compte **original et spécifique à votre site**.

Concernant le mot de passe, les bonnes pratiques en la matière s'appliquent ici plus que jamais. Choisissez un mot de passe **long** (au moins huit caractères), contenant de nombreux **caractères différents** (chiffres, lettres et autres symboles), et évitez les mots contenus dans le dictionnaire. Il est conseillé que ce mot de passe soit utilisé **uniquement pour ce site.**

Comment choisir un bon mot de passe lorsque vous créez un compte sur Internet ?
« Comment choisir un bon mot de passe lorsque vous créez un compte sur Internet ? » Cybermalveillance.gouv.fr : https://www.cybermalveillance.gouv.fr/tous-nos-contenus/actualites/comment-choisir-un-bon-mot-de-passe

C. Création des comptes utilisateurs

Une fois votre site créé, il faut désormais créer des **comptes distincts pour les différents utilisateurs** qui prendront part à son administration ou à la rédaction de contenu.

Partager votre compte administrateur avec les autres contributeurs peut sembler être une solution simple et pratique, mais cela pose de sérieux risques de sécurité ou même d'incidents involontaires si quelqu'un de non formé touche à des options qu'il ne devrait pas toucher… De plus, des accès propres à chacun faciliteront la gestion du contenu au sein de l'équipe du site.

Pour créer un nouveau compte depuis l'interface d'administration de votre site, cliquez sur **« Comptes«** puis sur **« Ajouter »** dans le menu latéral gauche.

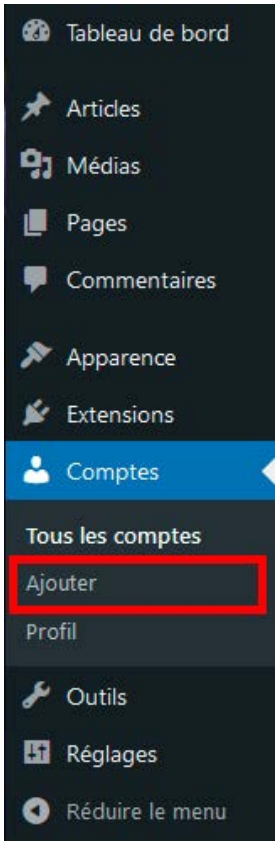


Fig. 2 Barre de menu latérale de l'interface d'administration. Encadré en rouge : le lien pour ajouter un nouveau compte d'utilisateur. © Richard Legrand

Lors du choix du **mot de passe**, les **conseils de sécurité** donnés précédemment s'appliquent également. Assurez-vous que l'interface désigne votre mot de passe comme **« fort »**. Si besoin, un mot de passe aléatoire fort est généré automatiquement lorsque vous commencez à ajouter un compte et peut être utilisé tel quel.

Extrait n°4.

LES PRINCIPALES FAILLES DE SÉCURITÉ DES APPLICATIONS WEB ET LEURS PARADES

Chaque faille peut être facilement exploitée par un hacker, même débutant, pour mener une attaque contre un site ou ses visiteurs. Il est pourtant tout aussi facile d'éviter ces failles grâce à des mesures de protection.

Les principales failles de sécurité des applications web et leurs parades

Nous allons dans cette partie du cours aborder trois failles de sécurité parmi les plus répandues dans les sites web. Chacune de ces failles peut être très facilement exploitée par un hacker, même débutant, pour mener une attaque contre un site ou ses visiteurs. Il est pourtant tout aussi facile, comme nous allons le voir, d'éviter ces failles grâce à des mesures de protection.

I. Injection SQL

A. Description

La grande majorité des sites web fonctionnent en conjonction avec une base de données, dans laquelle est stocké leur contenu dynamique ainsi que des informations confidentielles (comme les mots de passe ou les adresses mail des utilisateurs inscrits). Le langage SQL est le plus répandu pour préparer les requêtes, c'est-à-dire les instructions, envoyées à la base de données pour lire et écrire son contenu.

Dans certains cas, ces requêtes se font en fonction de données envoyées par un visiteur, par exemple un formulaire d'inscription qu'il aurait rempli et qui permettrait d'enregistrer son nom d'utilisateur et son mot de passe.

Une injection SQL a lieu lorsqu'un visiteur mal intentionné essaye d'envoyer du code SQL dans ces données, et que le site l'interprète comme une requête valide et l'exécute. **Cela offre au pirate un accès direct à la base de données** : il peut en extraire des informations confidentielles, les modifier ou encore les effacer.

Fig. 1 Schéma détaillant le fonctionnement d'une injection SQL. Un pirate, par l'intermédiaire d'un formulaire web, envoie des données malveillantes au serveur qui les transmet en tant que requête à la base de données. Cela permet au pirate de prendre le contrôle de la base de données et d'en récupérer ou modifier le contenu. © Richard Legrand

B. Protection

Nous l'avons déjà dit et nous le répétons : **il ne faut jamais faire confiance aux données émanant d'un utilisateur.** Au moment de transmettre les informations qu'il a envoyées au sein d'une requête vers la base de données, il faut s'assurer qu'elles ne puissent pas être interprétées comme du code SQL.

Cela peut être fait manuellement, en modifiant certains caractères qui permettent de délimiter le début ou la fin d'une requête. On dit alors que l'on **échappe** (*escape*) ces caractères.

Il est cependant conseillé de plutôt utiliser les mécanismes de sécurité prévus dans la plupart des langages de programmation, dans lesquels nous pouvons clairement **délimiter le code SQL normal prévu par le développeur, d'une part, et les variables saisies par l'utilisateur, d'autre part.**

II. Faille XSS (*Cross-Site Scripting*)

A. Description

Il est courant pour un site web d'offrir un espace où les visiteurs peuvent rédiger et déposer des messages qui seront ensuite affichés de façon publique. Pensez par exemple à un espace de commentaires sous un article de blog ; à un fil de discussion sur un forum ; ou aux avis de clients sur un site d'e-commerce…

Ces messages sont constitués d'un texte rédigé par le visiteur puis envoyé au site afin qu'il les stocke dans sa base de données, et enfin intégré au code HTML de la page web qui sera affiché à chaque fois que quelqu'un la consultera.

Mais que se passe-t-il si, plutôt que simplement envoyer du texte, un utilisateur malveillant insère **du code HTML et JavaScript** dans son message pour que ce code soit exécuté sur l'ordinateur de chaque nouveau visiteur ? Un code qui lui permettrait de rediriger le visiteur sur un autre site, ou lui demanderait d'installer un fichier…

C'est ce que l'on nomme une **faille XSS, pour *Cross-Site Scripting***, car elle permet à un hacker d'insérer des scripts directement dans vos pages web à travers votre site. C'est historiquement l'une des failles les plus courantes d'Internet.

Vous remarquerez qu'elle possède des similarités avec l'injection SQL : les deux profitent d'un formulaire pour envoyer du code malicieux afin qu'il soit exécuté. La différence est que l'injection SQL va chercher à faire exécuter son code directement dans le serveur du site pour prendre le contrôle de la base de données (côté serveur) tandis que la faille XSS va utiliser le site comme intermédiaire pour faire exécuter son code sur les ordinateurs des visiteurs (côté client).

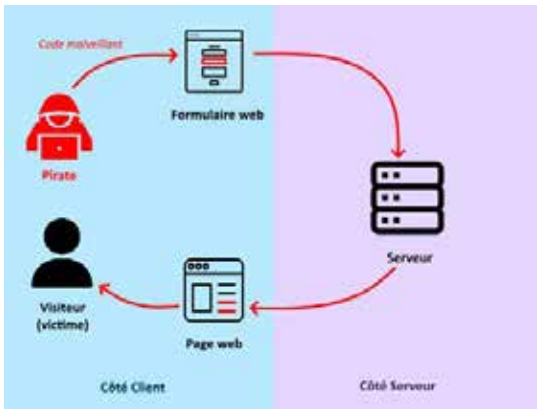


Fig. 2 Schéma détaillant le fonctionnement d'une faille XSS. Le pirate, par l'intermédiaire d'un formulaire web, poste du code malveillant sur le serveur, code qui sera ensuite affiché sur l'ordinateur des autres visiteurs de la page web. © Richard Legrand

B. Protection

Répétons tous en chœur : **il ne faut jamais faire confiance aux données émanant d'un utilisateur.**

Des protections doivent être mises en place dès que vous affichez du contenu rédigé par vos visiteurs, ou encore plus tôt, lorsque vous enregistrez ce contenu dans la base de données. Comme pour l'injection SQL, il faut s'assurer que **ce contenu ne puisse être interprété comme du code HTML ou JavaScript en retirant ou modifiant certains caractères clés**, comme les chevrons < et > qui servent à délimiter les balises HTML.

Si un certain degré de mise en forme de son texte est autorisé au visiteur (par exemple, on peut lui autoriser de mettre son texte en gras et en italique), on va alors **filtrer les balises HTML** contenues dans son message pour vérifier si elles sont autorisées ou dangereuses.

III. Falsification de requête intersites (CSRF)

A. Description

Une fois que vous vous connectez à un site web, celui-ci **se souvient de vous** pendant un certain temps : vous n'avez ainsi pas à entrer vos identifiants à nouveau à chaque fois que vous ouvrez une page. Cette « mémoire » du site est rendue possible grâce **à une session et à des cookies**, petits fichiers déposés sur votre ordinateur qui servent à vous identifier automatiquement. Ces fichiers disparaissent après un temps donné, mais dans l'intervalle, le site se « souviendra » de qui vous êtes.

Ainsi, imaginons qu'une internaute nommée Nina se connecte à sa banque en ligne pour regarder le solde de son compte, puis clique sur le lien « faire un virement ». Elle n'aura pas besoin de donner à nouveau son identifiant et son mot de passe pour confirmer le virement : le site de sa banque se « souvient » d'elle.

Imaginons maintenant qu'un internaute mal intentionné, que nous nommerons Michael, prépare un lien qui renvoie vers une page du site de la banque et permet de valider instantanément un virement d'argent vers son compte à lui. Il envoie ensuite ce lien à Nina par mail, inventant une excuse pour l'inciter à cliquer dessus. Si Nina s'est récemment connectée à sa banque en ligne et qu'elle clique sur le lien, le site se « souviendra » de ses identifiants et effectuera donc automatiquement le virement programmé… depuis le compte de Nina jusqu'au compte de Michael.

Cet exemple est bien entendu simplifié (rassurez-vous, les banques ont des mesures de protection contre cela), mais illustre bien le concept d'une attaque par **falsification de requête intersites** (*Cross-Site Request Forgery*, ou CSRF) : cela consiste à **faire effectuer une action à un autre utilisateur à son insu, profitant du fait qu'il soit connecté à un site.**

Comme vous le remarquez, cette attaque exploite non seulement l'aspect technique du site, mais également l'aspect psychologique de ses visiteurs. C'est donc une attaque qui met en jeu une dose d'**ingénierie sociale**.

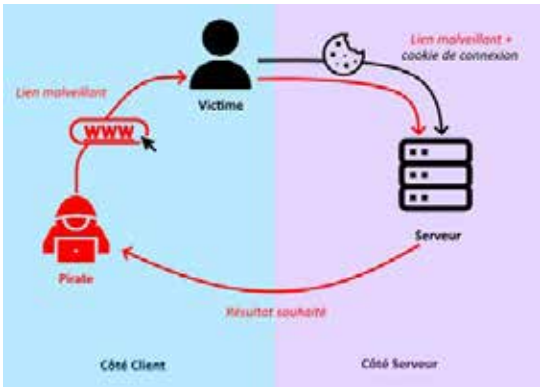


Fig. 3 Schéma détaillant le fonctionnement d'une falsification de requête intersite (CSRF). Le pirate envoie un lien malveillant à sa victime, déjà connectée à un site. Lorsqu'elle cliquera sur le lien, le site la reconnaîtra grâce à son cookie de connexion et lui fera effectuer l'action programmée par le pirate. © Richard Legrand

B. Parade

Pour éviter d'être victime d'une telle attaque en tant qu'internaute, assurez-vous de **la destination d'un lien** avant de cliquer dessus et déconnectez-vous de sites sensibles (banques, administrations…) une fois que vous avez cessé de les utiliser.

En tant que chef de projet web, assurez-vous que vos utilisateurs et administrateurs soient également **sensibilisés et formés** au sujet de cette attaque, ainsi que des autres basées sur l'ingénierie sociale.

En tant que concepteur ou développeur de site, des mesures techniques peuvent être aussi mises en place pour limiter les risques. On peut notamment intégrer dans les pages web **un message de confirmation** avant toute action importante, ou assurer **un délai d'expiration aux sessions de connexion des visiteurs.**

La manière la plus fiable est cependant d'insérer à tout échange de ce genre **un token** (petit code d'identification) secret et unique, transmis à l'ordinateur client au début de la transaction et vérifié à chaque étape pour s'assurer que l'identité du visiteur n'a pas changé en cours de route (comme cela a été le cas dans notre exemple entre Michael et Nina).

"La sécurité d'un site Web" - Mozilla.org
Pour aller plus loin au sujet des principales failles de sécurité des sites web : « La sécurité d'un site web » – Mozilla.org
https://developer.mozilla.org/fr/docs/Learn/Server-side/First_steps/Website_security

skill&you.

Envie d’en savoir plus ?
Ne tardez plus, planifiez votre rendez-vous.