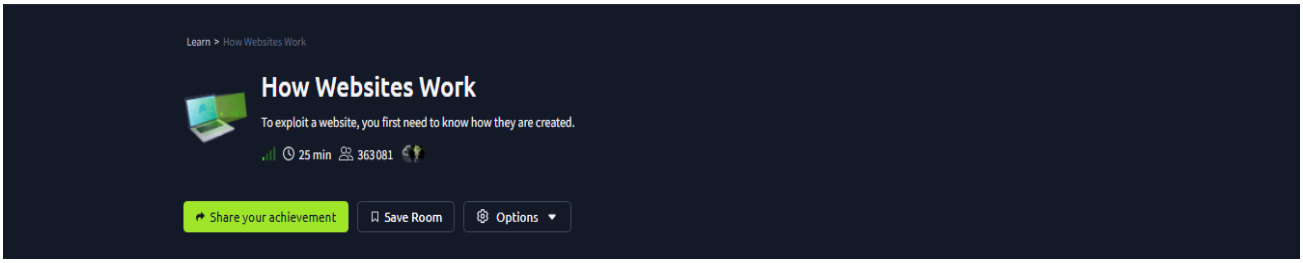


How Websites Work | 29 Septembre, 2025

Auteur : **GhostHunt3rx**
Email : **anonymousinconnu10@gmail.com**

Tâche 1 : Comment fonctionnent les sites web

À la fin de cette salle, vous saurez comment les sites web sont créés et vous découvrirez certaines questions fondamentales liées à la sécurité.

Lorsque vous visitez un site web, votre navigateur (comme Safari ou Google Chrome) envoie une requête à un serveur web pour demander des informations sur la page que vous visitez. Ce dernier répondra en envoyant des données que votre navigateur utilisera pour afficher la page. Un serveur web est simplement un ordinateur dédié, situé quelque part dans le monde, qui traite vos requêtes.

Un site web se compose de deux éléments principaux :

- **Front-end** (côté client) : la manière dont votre navigateur affiche un site web.
- **Back-end** (côté serveur) : un serveur qui traite votre demande et renvoie une réponse.

De nombreux autres processus interviennent lorsque votre navigateur envoie une requête à un serveur web, mais pour l'instant, il vous suffit de comprendre que vous envoyez une requête à un serveur et que celui-ci répond avec des données que votre navigateur utilise pour vous afficher les informations.

<u>Tâche 2</u> : HTML

Les sites web sont principalement créés à l'aide des langages suivants :

- HTML, pour créer des sites web et définir leur structure
- CSS, pour embellir les sites web en ajoutant des options de style
- JavaScript, pour implémenter des fonctionnalités complexes sur les pages à l'aide de l'interactivité

Le langage HTML (**H**yper**T**ext **M**arkup **L**anguage) est le langage dans lequel les sites web sont écrits. Les éléments (également appelés balises) sont les composants de base des pages HTML et indiquent au navigateur comment afficher le contenu. L'extrait de code ci-dessous montre un document HTML simple, dont la structure est la même pour tous les sites web :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>Example Heading</h1>
        <p>Example paragraph..</p>
    </body>
</html>
```

La structure HTML (comme le montre la capture d'écran) comporte les éléments suivants :

- La balise **<!DOCTYPE html>** définit que la page est un document HTML5. Cela facilite la normalisation entre les différents navigateurs et indique au navigateur d'utiliser HTML5 pour interpréter la page.
- L'élément **<html>** est l'élément racine de la page HTML - tous les autres éléments viennent après cet élément.
- L'élément **<head>** contient des informations sur la page (telles que le titre de la page).
- L'élément **<body>** définit le corps du document HTML ; seul le contenu à l'intérieur du corps est affiché dans le navigateur.

- L'élément **<h1>** définit un gros titre.
- L'élément **<p>** définit un paragraphe.
- Il existe de nombreux autres éléments (balises) utilisés à des fins différentes. Par exemple, il existe des balises pour les boutons (**<button>**), les images (**<img>**), les listes, et bien plus encore.

Les balises peuvent contenir des attributs tels que l'attribut class, qui peut être utilisé pour styliser un élément (par exemple, donner une couleur différente à la balise) **<p class="bold-text">**, ou l'attribut src, qui est utilisé sur les images pour spécifier l'emplacement d'une image : **<img src="img/cat.jpg">**. Un élément peut avoir plusieurs attributs, chacun ayant son propre objectif, par exemple **<p attribute1="value1" attribute2="value2">**.

Les éléments peuvent également avoir un attribut id (**<p id="example">**), qui est unique à l'élément. Contrairement à l'attribut class, où plusieurs éléments peuvent utiliser la même classe, un élément doit avoir des identifiants différents pour les identifier de manière unique. Les identifiants d'élément sont utilisés pour le style et pour les identifier par JavaScript.

Vous pouvez afficher le code HTML de n'importe quel site web en cliquant avec le bouton droit de la souris et en sélectionnant « Afficher la source de la page » (Chrome) / « Afficher la source de la page » (Safari).

Tâche 3 : JavaScript

JavaScript (JS) est l'un des langages de programmation les plus populaires au monde. Il permet de rendre les pages interactives. Le langage HTML sert à créer la structure et le contenu d'un site web, tandis que JavaScript sert à contrôler les fonctionnalités des pages web. Sans JavaScript, une page ne comporterait aucun élément interactif et serait toujours statique. JS peut mettre à jour la page de manière dynamique et en temps réel, ce qui permet de modifier le style d'un bouton lorsqu'un événement particulier se produit sur la page (par exemple, lorsqu'un utilisateur clique sur un bouton) ou d'afficher des animations animées.

JavaScript est ajouté dans le code source de la page et peut être chargé dans des balises **<script>** ou inclus à distance avec l'attribut src : **<script src="/location/of/javascript_file.js"></script>**

Le code JavaScript suivant recherche un élément HTML sur la page avec l'identifiant « demo » et modifie le contenu de l'élément en « Hack the Planet » : **document.getElementById("demo").innerHTML = "Hack the Planet";**

Les éléments HTML peuvent également avoir des événements, tels que « onclick » ou « onhover », qui exécutent JavaScript lorsque l'événement se produit. Le code suivant remplace le texte de l'élément dont l'ID est « demo » par « Button Clicked » : **<button onclick='document.getElementById("demo").innerHTML = "Button Clicked";'>**Click Me!</button> - Les événements onclick peuvent également être définis à l'intérieur des balises de script JavaScript, et non directement sur les éléments.

Tâche 4 : Exposition des données sensibles

L'exposition de données sensibles se produit lorsqu'un site web ne protège pas correctement (ou ne supprime pas) les informations sensibles en clair destinées à l'utilisateur final ; elle se trouve généralement dans le code source frontal d'un site.

Nous savons désormais que les sites web sont construits à l'aide de nombreux éléments HTML (balises), que nous pouvons tous voir simplement en « affichant la source de la page ». Un développeur de site web peut avoir oublié de supprimer les identifiants de connexion, les liens cachés vers des parties privées du site web ou d'autres données sensibles affichées en HTML ou JavaScript.

Les informations sensibles peuvent être potentiellement exploitées pour faciliter l'accès d'un pirate à différentes parties d'une application web. Par exemple, il peut y avoir des commentaires HTML contenant des identifiants de connexion temporaires. Si vous consultez le code source de la page et que vous les trouvez, vous pouvez utiliser ces identifiants pour vous connecter à d'autres parties de l'application (ou pire, pour accéder à d'autres composants backend du site).

Lorsque vous évaluez la sécurité d'une application web, l'une des premières choses à faire est d'examiner le code source de la page pour voir si vous pouvez trouver des identifiants de connexion exposés ou des liens cachés.

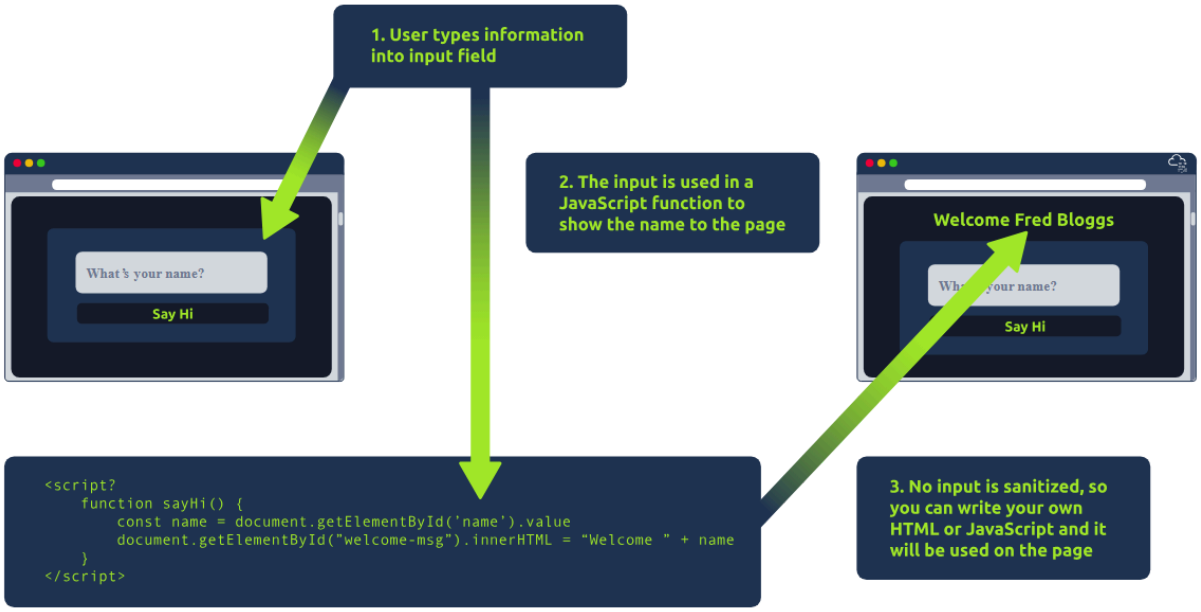
```
<!DOCTYPE html>
<html>
    <head>
        <title>Fake Website</title>
    </head>
    <body>
       <form>
          <input type='text' name='username'>
          <input type='password' name='password'>
          <button>Login</button>
          <!-- TODO: remove test credentials admin:password123 -->
        </form>
    </body>
</html>
```

Tâche 5 : Injection HTML

L'injection HTML est une vulnérabilité qui se produit lorsque des données utilisateur non filtrées sont affichées sur la page. Si un site web ne parvient pas à nettoyer les données utilisateur (filtrer tout texte « malveillant » saisi par un utilisateur sur un site web) et que ces données sont utilisées sur la page, un pirate peut injecter du code HTML dans un site web vulnérable.

Le nettoyage des données saisies est très important pour assurer la sécurité d'un site web, car les informations saisies par un utilisateur sur un site web sont souvent utilisées dans d'autres fonctionnalités frontales et dorsales. Une vulnérabilité que vous explorerez dans un autre laboratoire est l'injection de base de données, où vous pouvez manipuler une requête de recherche dans une base de données pour vous connecter en tant qu'autre utilisateur en contrôlant les données saisies qui sont directement utilisées dans la requête. Mais pour l'instant, concentrons-nous sur l'injection HTML (qui se fait côté client).

Lorsqu'un utilisateur contrôle l'affichage de ses entrées, il peut soumettre du code HTML (ou JavaScript), et le navigateur l'utilisera sur la page, ce qui lui permettra de contrôler l'apparence et les fonctionnalités de la page.



L'image ci-dessus montre comment un formulaire affiche du texte sur la page. Tout ce que l'utilisateur saisit dans le champ « Quel est votre nom » est transmis à une fonction JavaScript et affiché sur la page. Cela signifie que si l'utilisateur ajoute son propre code HTML ou JavaScript dans le champ, celui-ci est utilisé dans la fonction sayHi et ajouté à la page. Vous pouvez donc ajouter votre propre code HTML (tel qu'une balise <h1>) et votre saisie sera affichée en HTML pur.

La règle générale est de ne jamais faire confiance aux entrées des utilisateurs. Pour éviter les entrées malveillantes, le développeur du site web doit nettoyer tout ce que l'utilisateur saisit avant de l'utiliser dans la fonction JavaScript ; dans ce cas, le développeur pourrait supprimer toutes les balises HTML.