Université Abdelmalek Essaadi

Ecole Nationale des Sciences Appliquées Al-Hoceima

Première Année Transformation digitale et intelligence artificielle

Recherche sur le protocole http, https, les outils de développement chrome et la communauté OWASP:

Réalisée par :

- Mohammed ACHBAB

Encadré par :

- MOHAMED CHERRADI

Table des matières

1. Protocole HTTP:

1.1. protocole http en générale :

En générale, le protocole http sert à la communication entre le navigateur(le client) et le serveur contenant les sites web cette communication ce fait sous formes de requête.

Autrement dit, c'est le protocole utilisé pour les transactions Web. Il permet d'établir un dialogue entre un client (généralement, un navigateur) et un serveur Web.
C'est un protocole sans état (Le client émet une requête, le serveur l'analyse, émet une réponse et C'est terminé).

1.2. exemple d'une requête http (avec la méthode GET) :

GET <www.nom_du_site.com> <version du protocole http>

----------en-têtes de requête http :----------

User-Agent : <moteur de recherche (google/Mozilla)>

Accept : <ressources attendues (text/html…)>

Accept langage : <langue attendue (fr-FR/en-US…)>

La première version du protocole http est http0.9 donc quelles sont les améliorations que la version 1.0 a apporter ?

1.3. Difference entre les protocoles http/0 ,http/1, et http/2 :

La version HTTP/0.9 :

- Elle était très simple et ne supportait que la méthode GET pour récupérer des documents.
- Elle ne supportait pas les en-têtes de requête HTTP, ni les codes d'état de réponse
- Elle était conçue pour des usages très simples et ne convenait pas à des cas d'utilisation avancés ou à grande échelle

La version HTTP/1 :

- Introduite en 1996, cette version a apporté plusieurs améliorations par rapport à la version 0.9.
- Elle a introduit la notion de méthodes HTTP autres que GET, telles que POST, PUT, DELETE, etc.
- Elle a introduit des en-têtes de requête HTTP, permettant de spécifier des informations supplémentaires dans les requêtes.

La version http/2 :

- http/2 est plus rapide et performant que HTTP/1.1
- HTTP/2 utilise une couche de trame binaire pour encapsuler les messages en format binaire (rendre les messages(requêtes) sous format binaire), améliorant l'efficacité par rapport à HTTP/1.1 qui garde les requêtes et réponses en texte brut, offrant ainsi des fonctionnalités et optimisations de performances telles que le multiplexage des requêtes, la compression des en-têtes et l'optimisation du transport des sémantiques http :
    - Multiplexage des Requêtes: permet d'envoyer et de recevoir plusieurs requêtes simultanément sur une seule connexion.
    - Compression des En-têtes : réduit la taille des en-têtes HTTP envoyés sur le réseau. En compressant ces en-têtes, on économise de la bande passante et on diminue le temps de transfert des données, ce qui contribue à une meilleure performance globale du protocole.
    - Optimisation du Transport des Sémantiques HTTP: vise à améliorer la manière dont les données sont échangées entre le client et le serveur.
- Multiplexage : HTTP/2 permet le multiplexage des requêtes, ce qui signifie que plusieurs requêtes peuvent être traitées simultanément sur une seule connexion, contrairement à HTTP/1.1 où chaque requête nécessite une nouvelle connexion

1.4. les différentes méthodes utilisés dans le protocole http:

- GET :
- Utilisé pour demander des données à partir d'une ressource spécifiée.
- Les données envoyées sont incluses dans l'URL sous forme de chaîne de requête.
- Normalement utilisé pour lire des données et ne doit pas avoir d'effets secondaires sur le serveur.
- POST :
- Utilisé pour soumettre des données à être traitées à partir de la ressource spécifiée.
- Les données envoyées sont incluses dans le corps de la requête HTTP.
- Normalement utilisé pour envoyer des données à un serveur pour créer ou mettre à jour une ressource.
- PUT :
- Utilisé pour envoyer des données à une ressource spécifiée, remplaçant entièrement ou partiellement les données existantes.
- Les données envoyées sont incluses dans le corps de la requête HTTP.
- Souvent utilisé pour mettre à jour une ressource existante ou pour créer une nouvelle ressource si elle n'existe pas déjà.
- DELETE :
- Utilisé pour supprimer la ressource spécifiée.
- Aucune donnée n'est généralement envoyée avec la requête, bien que certaines implémentations puissent accepter des données dans le corps de la requête pour fournir des informations supplémentaires sur la ressource à supprimer.
- PATCH :
- Utilisé pour appliquer des modifications partielles à une ressource.
- Le corps de la requête contient un ensemble d'instructions décrivant les modifications à appliquer à la ressource.
- OPTIONS :

- Utilisé pour obtenir les options de communication disponibles pour la ressource cible, telles que les méthodes HTTP supportées.
- Utile pour découvrir les fonctionnalités et les exigences de sécurité d'un serveur.

Ces différentes méthodes permettent aux clients et aux serveurs HTTP d'interagir de manière spécifique avec les ressources sur le Web, fournissant des mécanismes flexibles pour récupérer, soumettre, mettre à jour ou supprimer des données en fonction des besoins de l'application.

2. Protocole https :

2.1. Différence entre http et https :

- HTTPS est plus sécurisé que http, il utilise un protocole sécurisé pour le transfert de données, assurant ainsi la confidentialité et l'intégrité des données échangées entre le serveur et le navigateur.
- Au niveau de sécurité, Les données sont chiffrées à l'aide de protocoles de sécurité tels que SSL (Secure Sockets Layer) ou TLS (Transport Layer Security) avant d'être transférées sur Internet. Cela garantit que même si quelqu'un intercepte les données, il ne pourra pas les comprendre car elles sont chiffrées.
- Par défaut https utilise le port 443, alors que http utilise le port 80

2.2. Différence entre https/1.0, https/1.1 et https/2 :

- HTTPS/1.0 : La version HTTPS/1.0 n'est pas couramment utilisée, étant remplacée rapidement par des versions ultérieures, car elle nécessite d'établir une nouvelle connexion SSL pour chaque requête.
- HTTPS/1.1 :
  - Il a introduit des fonctionnalités telles que la persistance de la connexion (keep-alive), qui permet de réutiliser une seule connexion TCP pour plusieurs requêtes, réduisant ainsi les retards.
  - HTTPS/1.1 a également inclus des améliorations pour la négociation de la sécurité et la gestion des caches.
- HTTPS/2 :
  - HTTPS/2 est la dernière version du protocole HTTPS et apporte des améliorations significatives en termes de performances et d'efficacité par rapport à HTTPS/1.1.
  - Il utilise une seule connexion TCP multiplexée pour gérer plusieurs flux de données simultanément, ce qui réduit la surcharge des connexions et améliore les performances.
  - HTTPS/2 prend en charge la compression des en-têtes HTTP, ce qui réduit la surcharge du réseau.
  - Il introduit également la priorisation des flux, permettant aux clients de spécifier l'ordre de priorité des ressources à télécharger.

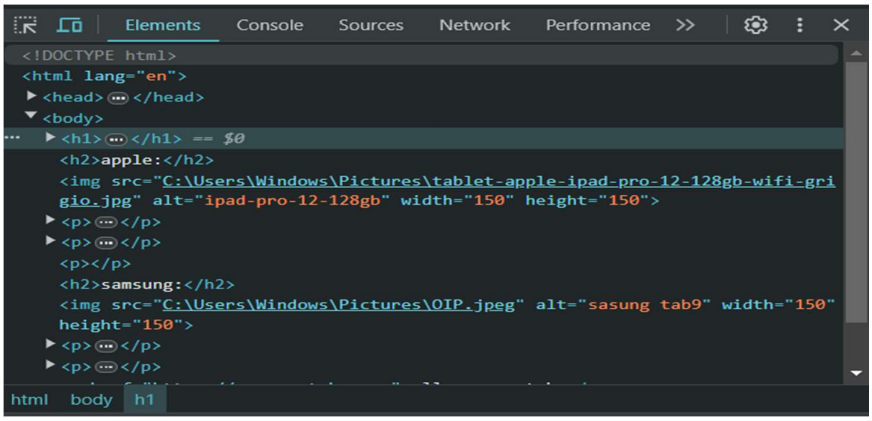
3. Outils de développement chrome (chrome devtools) :

Les Chrome DevTools sont un ensemble d'outils de développement intégrés dans le navigateur Google Chrome. Ils offrent un large éventail de fonctionnalités destinées à aider les développeurs web à inspecter, tester, déboguer et optimiser leurs sites et applications web.
Autrement dit, Les outils de développement Chrome sont une suite d'outils intégrés à Google Chrome qui facilitent le développement et le débogage de sites web.
Il y a plusieurs méthodes pour accéder à ces outils de développements, la plus fréquente est clic droit sur la page web puis on clique sur inspecter.

Après cette étape il nous affiche ces outils.

Le bloc "Eléments" :

Le bloc "Eléments" dans les DevTools Chrome est un outil qui contient le code html du site, il permet d'inspecter et de modifier le HTML en direct sur une page web. Il permet d'accéder à l'arborescence HTML de la page et aux styles CSS appliqués à ces éléments. On peut également effectuer des modifications en direct sur le HTML et voir les résultats immédiatement sur la page web en cliquant sur la portion du code voulue puis on La changeant.



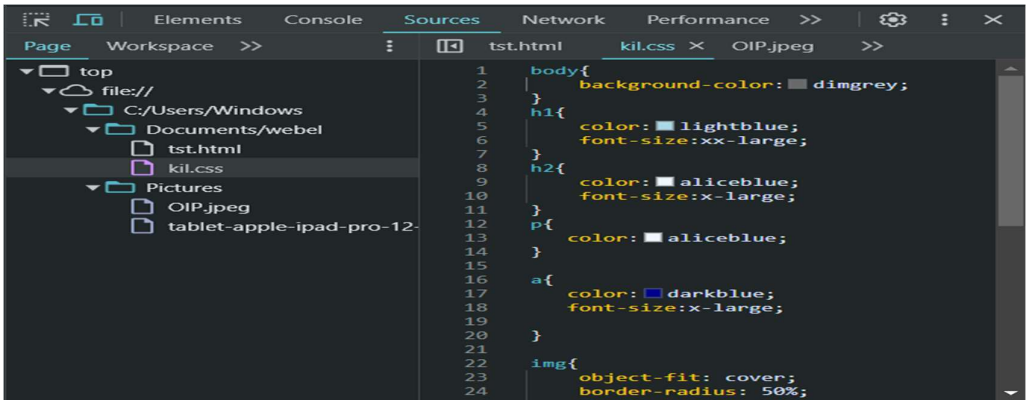
Le bloc "Console" :

C'est un outil qui permet d'examiner et de modifier les messages JavaScript et les événements du navigateur. On peut utiliser la console pour afficher des messages de débogage, exécuter des commandes JavaScript, inspecter les variables et les objets JavaScript, et plus encore.

Le bloc "Sources" :

C'est un outil qui permet d'accéder et de modifier les fichiers sources d'un site web. Il offre des fonctionnalités avancées pour l'édition du code, le débogage JavaScript, la mise en place de points d'arrêt pour le débogage, la navigation dans les fichiers sources, et bien plus encore. Cet onglet est essentiel pour les développeurs web car il leur permet de travailler directement sur le code source du site et d'effectuer des modifications en profondeur pour améliorer la performance et la fonctionnalité du site

Par exemple :



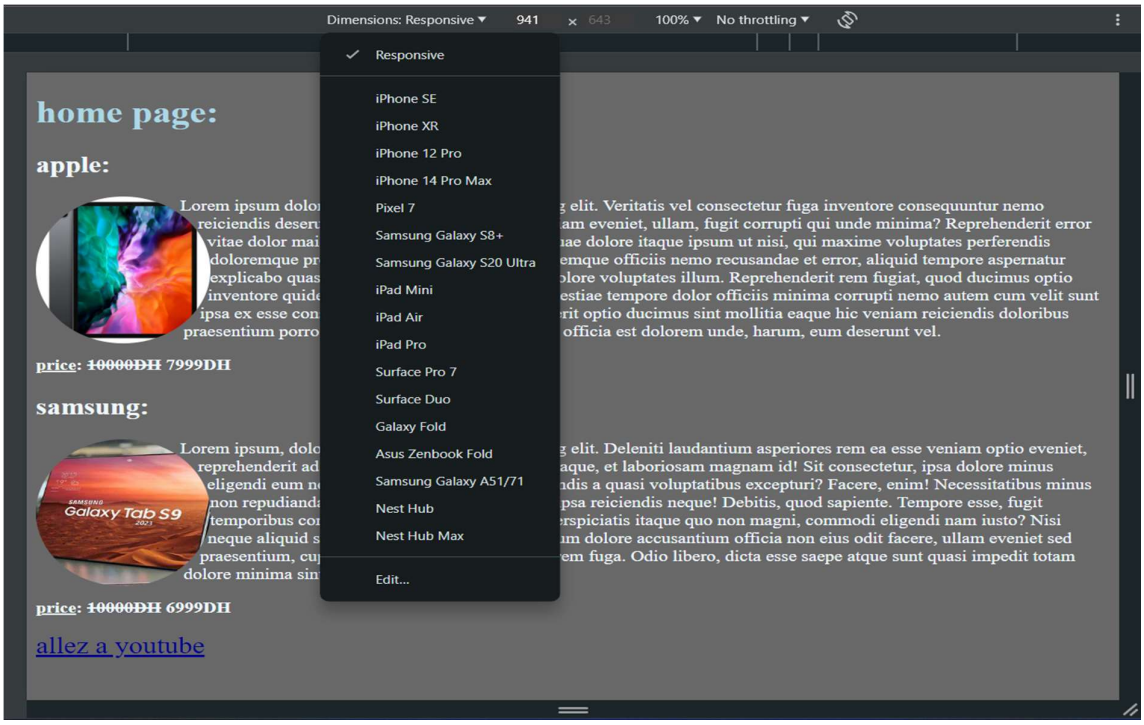
Dans cette exemple source nous a afficher les diffèrent fichiers sources utilisée dans ce site

Le bloc "Network" :

Il permet d'analyser et de déboguer les appels réseau effectués par une page web. Il affiche les requêtes HTTP effectuées par la page, les réponses reçues du serveur, les temps de chargement des ressources, les codes de statut des requêtes, et bien plus encore. Cet outil est essentiel pour comprendre les performances d'un site web, identifier les problèmes de chargement des ressources, optimiser la vitesse de chargement et diagnostiquer les problèmes liés aux appels réseau

Autres outils :

On peut aussi tester l'apparence du site dans les diffèrent appareils par la barre qui s'affiche au-dessus de la page :



➢ L'outil "Performance" :

Permet d'analyser en détail les performances des animations et du code JavaScript d'une page web. Cet outil offre la possibilité d'identifier les problèmes potentiels liés aux performances, de visualiser le temps pris par chaque fonction JavaScript, d'analyser les appels de fonctions, et de diagnostiquer les éventuels problèmes de performance.

4. OWASP

4.1. Définition :

- L'OWASP (Open Web Application Security Project) est une communauté internationale dédiée à la sécurité des applications web, Fondée en 2001.
- l'OWASP fournit des ressources gratuites et ouvertes à tous pour améliorer la sécurité des logiciels.
- Ces buts et objectifs :

➔Sensibiliser : Sensibiliser les entreprises, les développeurs et le grand public aux problèmes de sécurité des applications web.

➔Éduquer : Fournir des ressources éducatives telles que des guides, des outils, des projets, et des formations pour aider à comprendre et à résoudre les problèmes de sécurité des applications web.

➔Fournir des outils : Développer et distribuer des outils open source pour évaluer et améliorer la sécurité des applications web (OWASP ZAP (Zed Attack Proxy)).

➔Communauté : Encourager la collaboration entre professionnels de la sécurité et développeurs pour promouvoir les meilleures pratiques en matière de sécurité des applications.

4.2. Les 5 premières vulnérabilités à partir de OWASP Top Ten (année 2017):

Les dix vulnérabilités les plus courantes et les plus lourdes de conséquences qui apparaissent dans les applications web sont les suivantes :

- Injection (exemple SQL) : Les attaques par injection SQL se produisent lorsque des données non fiables sont insérées dans des requêtes SQL (injectés), permettant aux attaquants d'exécuter des commandes malveillantes sur la base de données.
- Authentification brisée (Broken Authentication) : Cela se produit lorsque les mécanismes d'authentification ne sont pas correctement mis en œuvre, permettant aux attaquants de compromettre les comptes d'utilisateurs, d'accéder à des informations sensibles ou de prendre le contrôle du système puisqu'il a réussi à accéder à la base de donnée.
- Exposition de données sensibles : Lorsque des données sensibles sont stockées ou transmises de manière non sécurisée, elles peuvent être exposées aux attaquants, ce qui baisse ainsi la confidentialité des informations. Cette vulnérabilités a pris le nom « Les défaillances cryptographiques » et a montée à la deuxième place en 2021
- Entités externes XML (XXE) : Les attaques XXE exploitent les fonctionnalités XML pour accéder à des ressources système sensibles, exécuter du code à distance ou provoquer un déni de service.
- La mauvaise configuration de la sécurité (Broken Access Control) : faille de sécurité critique dans les logiciels. Elle est devenue la première vulnérabilité critique en 2021 selon l'OWASP. Cette faille peut permettre à un attaquant de contrôler à partir d'un compte un autre compte. Autrement dit ,elle permet à l'attaquant de contourner les contrôles d'accès établis, lui donnant ainsi la possibilité de visualiser, modifier ou supprimer du contenu, exécuter des fonctions non autorisées, aussi prendre le contrôle complet d'un site web

4.3. Différence de classement des Top 10 vulnérabilités de OWASP entre les années 2017 et 2021

Il y a trois nouvelles catégories, quatre catégories avec des changements de nommage et de portée, et une certaine consolidation dans le Top 10 pour 2021 :